Generative Modeling in Large-scale Dynamical Systems

François Rozet

advised by

Gilles Louppe

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

to the

Department of Electrical Engineering and Computer Science
University of Liège

November 28, 2025

Les questions les plus importantes de la vie ne sont, pour la plupart, que des problèmes de probabilitié. On peut même dire que presque toutes nos connaissances ne sont que probables; et dans le petit nombre des choses que nous pouvons savoir avec certitude, les principaux moyens de parvenir à la vérité – l'induction et l'analogie – se fondent sur les probabilités, de sorte que le système entier des connaissances humaines se rattache à la théorie des probabilités.

— Pierre-Simon Laplace (1820)

Jury

Marc Bocquet
Professor at École des Ponts

Pierre Geurts
Professor at the Univerity of Liège

Philipp Hennig
Professor at the University of Tübingen

Gilles Louppe (Advisor)
Professor at the Univerity of Liège

Antoine Wehenkel
Research Scientist at Apple Inc.

Louis Wehenkel (President)
Professor at the Univerity of Liège

Contents

I
Prologue

1 Introduction

> *Une intelligence qui, pour un instant donné, connaîtrait toutes les forces dont la nature est animée, et la situation respective des êtres qui la composent, si d'ailleurs elle était assez vaste pour soumettre ces données à l'analyse, embrasserait dans la même formule les mouvements des plus grands corps de l'univers et ceux du plus léger atome; rien ne serait incertain pour elle, et l'avenir, comme le passé, serait présent à ses yeux.*
>
> — Pierre-Simon Laplace (1840)

Laplace's demon is a hypothetical superintelligent being that knows everything about the state of our universe, from the largest star to the smallest particle, at a single moment in time. One can imagine that the demon possesses a kind of photograph of our universe, recording not just light from one point of view, but all physical properties of all objects in all directions. Then, under the assumption that our universe is a colossal deterministic machine, nothing from ancient past to distant future is uncertain for the demon. For us, mortal humans, the picture is very different. First, we are not superintelligent. We are limited by the power of our brains and, since 1945, computers. Even if we had access to as much information as Laplace's demon, we would not be able to process it on a human timescale. Second, unlike the demon, we don't know the machinery behind our universe. Through the scientific method, we make, falsify, and replace models, but they remain incomplete. In fact, to this day, it is still debated whether our universe is inherently deterministic or stochastic [2–4]. If it is stochastic, no amount of information would be enough to predict the future with certainty. Third, and perhaps most importantly, we are rarely certain about the state of a system, let alone of our universe. Our instruments, which enable us to probe systems and gather observations, are never perfect and always carry some level of measurement error. Even if we could develop more precise instruments, Heisenberg's uncertainty principle [5] states that there is a fundamental limit to the precision with which some physical properties, such as position and momentum, can be known simultaneously. All these limitations make human beings inherently uncertain.

All models are wrong, some are useful.

— George Box (1976)

Despite our doubts, we are and have been obsessed with predicting the future for millennia. From the Babylonians and Mayans to the Chinese and Greeks, we have tried to predict the movements of planets, the changes in weather, the apparition of natural disasters, and much more. Our ancestors, who were more vulnerable to their environment than we are today, predicted out of necessity: they had to protect, feed, and shelter their families. Unfortunately, while ancient astronomers made remarkably acute predictions of planetary and stellar motions, weather forecasting remained wildly speculative for centuries. It is only by the end of the Renaissance that the first instruments to measure temperature and atmospheric pressure, the thermometer and barometer, were invented by Italian physicists Galileo Galilei and Evangelista Torricelli. These early instruments, and those that followed, enabled individuals to make and record surface measurements across the globe. The later emergence of telegraph networks, during the nineteenth century,

Le doute n'est pas un état bien agréable, mais l'assurance est un état ridicule.

— Voltaire (1770)

made it possible to share and compile observations routinely. Crude weather maps were drawn and surface wind patterns started to be identified and studied. At the turn of the twentieth century, Abbe [6] and Bjerknes [7] proposed that the laws of physics could be used to model the atmospheric dynamics and forecast the weather. This proposition was audacious at that time, as there were few routine observations of the atmosphere, limited understanding of meteorological mechanisms, and no computers. Since, there has been continued investment and advances in the observational, theoretical and computational aspects of geosciences, enabling ever more reliable weather forecasts. Meteorological instruments now include polar-orbiting and geostationary satellites, ocean buoys, cloud radars, and surface weather stations, among many others. This network of instruments, illustrated in Figure 1.1, is known as the Global Observing System [8] and is in constant expansion.

However, better and higher-resolution instruments imply greater sensitivity to physical and chemical processes in the oceans and atmosphere, which in turn increases the complexity of the theoretical models required to interpret their measurements. The growing number of instruments also introduces new challenges in terms of quantity of collected data. Most notably, the use of satellite data has increased exponentially over the last two decades, becoming the largest contributor to forecast quality [9] and pushing current numerical weather prediction (NWP) methods to their limits. In fact, an overwhelming majority (90 to 95 %) of the available satellite data is unexploited by operational NWP centers [10], including satellite channels in the visible and infrared spectrum, despite providing a wealth of information on atmospheric clouds [11, 12]. These observations stand largely unassimilated because (a) efficient models of their measurement processes are still lacking, and (b) scaling current pipelines to ingest their volume remains computationally intractable.

Undeterred by these challenges, meteorologists across the globe continue their work towards more accurate weather prediction, for its impact is among the greatest of any field in the physical sciences [10]. However, many now argue that, instead of investing solely in new instruments and increasingly sophisticated models, NWP should adopt a new computational paradigm; one that is more efficient, takes full advantage of modern hardware, and learns from the vast amounts of available data. This is where machine learning and, more specifically, deep learning come into play. Amid this abundance of data, where experts struggle to extract patterns and leverage domain knowledge, machines can learn efficient models automatically. Over the past decade, this has proven true for an increasingly broad spectrum of tasks, including language modeling [14–16], voice processing [17–19], image synthesis [20–22], autonomous agents [23–27], and

Machines take me by surprise with great frequency.

— Alan Turing (1950)

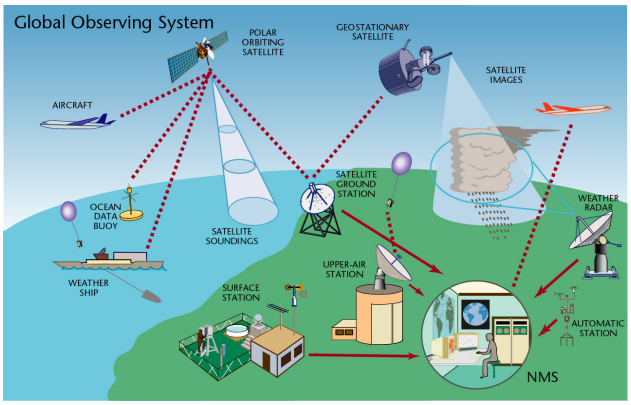


Figure 1.1. Illustration of the components of the Global Observing System (GOS) by the World Meteorological Organization [8].

numerous scientific applications [28–33]. Following this success, a wave of research has sought to enhance or replace NWP components with deep learning models [34–45], some of which have already been integrated into operational pipelines [46, 47].

Still, unless one is akin to Laplace's demon, prediction remains fundamentally riddled with uncertainty. While there has been a long history of work to integrate deep learning models in the natural sciences, it is only recently that practitioners have started using deep *probabilistic* models. These models explicitly take uncertainty into account and learn the non-deterministic relationships between the quantities of interest. Among the first deep probabilistic models were probability density estimators, including density ratio estimators [48–64] and normalizing flows [65–73]. One area where density estimators became the gold standard is simulation-based inference [74], which is now recognized as a valuable extension to the statistical toolbox used in particle physics [75].

Unfortunately, density estimators have classically been restricted to low-dimensional settings, making their application to high-dimensional problems, such as those encountered in earth sciences, challenging. Furthermore, while estimating the probability density has many use cases, drawing samples from a target distribution is often the objective of probabilistic approaches. In response to these needs, a new class of deep probabilistic models has emerged, that can efficiently generate samples from high-dimensional distributions. These *generative* models notably include variational auto-encoders [76, 77], generative adversarial networks [78], and, most recently, diffusion models [79, 80], also known as score-based generative models [81, 82], flow models [83, 84], or stochastic interpolants [85]. Since their introduction, generative models have received widespread adoption in the context of image, video, and audio synthesis, and their success is now spreading to all areas of natural sciences.

1.1 Research Question

Motivated by the recent wave of research to solve scientific problems with deep probabilistic models, and the remaining obstacles to tackle high-dimensional settings, this thesis focuses on developing and applying generative models to large-scale inference problems in physics, particularly in systems whose state evolves through time, known as dynamical systems. The rapid evolution of deep learning and the modern abundance of scientific data make this objective not only timely, but a unique opportunity for technological progress.

I will not define time, space, place and motion, as being well known to all.

— Isaac Newton (1687)

In pursuit of this objective, this thesis explores several aspects of probabilistic modeling and dynamical systems, including state estimation, forecasting, reduced-order modeling, and learning from corrupted observations. Although motivated and developed in the context of scientific applications, the methodological contributions extend to the broader field of generative modeling.

1.2 Outline

Immediately following this introduction, we present the main contributions of this thesis, which are based on a series of peer-reviewed publications. We choose not to include an original primer on dynamical systems or deep probabilistic modeling as part of this dissertation. These topics are thoroughly covered in the literature through numerous books and reviews, and our own attempts could not match their clarity and completeness. We notably recommend two excellent reviews by Carrassi et al. [87] and Lai et al. [88] on data assimilation and diffusion models, respectively. The background concepts and relevant literature are nevertheless discussed within the respective chapter of each contribution.

Opening the scientific part of this dissertation, Chapter 2 investigates the application of diffusion models [79, 80] to the problem of state estimation in dynamical systems. This work establishes diffusion models as a promising and versatile alternative to classical data assimilation [87, 89–92] methods. Chapter 3 further demonstrates that the method presented in Chapter 2, named score-based data assimilation (SDA), scales effectively to near-operational-scale systems. However, the data SDA relies on for training – fully observed state sequences – is rarely accessible. In practice, only incomplete and noisy observations of the states are available, especially in earth and space sciences where the systems of interest can only be probed superficially. Chapter 4 addresses this limitation in the broader context of learning a prior distribution parameterized by a diffusion model when only partial and noisy observations are available. The proposed method is an adaptation of the expectation-maximization [93–97] algorithm tailored to diffusion models. Another limitation of SDA is its computational cost, which becomes prohibitive for long trajectories and high-dimensional states. In the context of image and video generation, this computational obstacle has been mitigated by generating in the latent space of an autoencoder rather than in pixel space. In Chapter 5, we investigate whether a similar strategy can be effectively applied to the emulation of dynamical systems, and at what cost. Our results indicate that latent modeling is a promising approach; one that is not only more efficient, but can also improve accuracy compared to physics-space models. Encouraged by these findings, we combine the latent modeling approach with the SDA framework in Chapter 6, obtaining promising results on operational-scale atmospheric data assimilation tasks.

Finally, Chapter 7 reflects upon the contributions of this thesis and discusses possible future developments in deep probabilistic modeling and its applications to dynamical systems.

1.3 Publications

The scientific content of this dissertation is exclusively borrowed from the following peer-reviewed publications, for which I played the role of lead author or project instigator.

François Rozet and Gilles Louppe. “Score-based Data Assimilation”. In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

François Rozet and Gilles Louppe. “Score-based Data Assimilation for a Two-Layer Quasi-Geostrophic Model”. In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2023.

François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. “Learning Diffusion Priors from Observations by Expectation Maximization”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

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”. In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

Gérôme Andry, Sacha Lewin, François Rozet, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, and Gilles Louppe. “Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation”. In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2025.

Throughout the pursuit of my degree, I also had the chance to take part and contribute to fruitful collaborations. The following list of publications, which are not featured in

this thesis, stemmed from these collaborations.

Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. "A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful". In *Transactions on Machine Learning Research* (2022).

Arnaud Delaunoy, Joeri Hermans, François Rozet, Antoine Wehenkel, and Gilles Louppe. "Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

Malavika Vasist, François Rozet, Olivier Absil, Paul Mollière, Evert Nasedkin, and Gilles Louppe. "Neural posterior estimation for exoplanetary atmospheric retrieval". In *Astronomy & Astrophysics* 672 (2023).

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno R. Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. "The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

Thomas Savary, François Rozet, and Gilles Louppe. "Training-Free Data Assimilation with GenCast". In *Tackling Climate Change with Machine Learning Workshop (NeurIPS)*. 2025.

Rudy Morel, Francesco Pio Ramunno, Jeff Shen, Alberto Bietti, Kyunghyun Cho, Miles Cranmer, Siavash Golkar, Olexandr Gugnin, Geraud Krawezik, Tanya Marwah, Michael McCabe, Lucas Thibaut Meyer, Payel Mukhopadhyay, Ruben Ohana, Liam Holden Parker, Helen Qu, François Rozet, K. D. Leka, Francois Lanusse, David Fouhey, and Shirley Ho. "Predicting partially observable dynamical systems via diffusion models with a multiscale inference scheme". In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Regaldo-Saint Blancard, François Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, François Lanusse, Rudy Morel, Ruben Ohana, Liam Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. "Walrus: A Cross-Domain Foundation Model for Continuum Dynamics". 2025.

Finally, over the past years, I have devoted a significant part of my time to free and open-source software. I consider the following Python packages, which I created and maintained during my degree, to be among its contributions.

François Rozet, Arnaud Delaunoy, Benjamin Miller, et al. "LAMPE: Likelihood-free amortized posterior estimation". 2021.
https://pypi.org/project/lampe

François Rozet et al. "Zuko: Normalizing flows in PyTorch". 2022.
https://pypi.org/project/zuko

François Rozet. "Azula: Diffusion models in PyTorch". 2024.
https://pypi.org/project/azula

References

[1] Pierre Simon Laplace. "Essai philosophique sur les probabilités". Paris Bachelier, 1840.

[2] J. S. Bell. "On the Einstein Podolsky Rosen paradox". In *Physics Physique Fizika* 1.3 (1964).

[3] Marco Genovese. "Research on hidden variable theories: A review of recent progresses". In *Physics Reports* 413.6 (2005).

[4] Jan-Åke Larsson. "Loopholes in Bell inequality tests of local realism". In *Journal of Physics A: Mathematical and Theoretical* 47.42 (2014).

[5] W. Heisenberg. "Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik". In *Zeitschrift für Physik* 43.3 (1927).

[6] Cleveland Abbe. "The Physical Basis of Long-Range Weather Forecasts". In *Monthly Weather Review* 29.12 (1901).

[7] Vilhelm Bjerknes. "The problem of weather prediction, considered from the viewpoints of mechanics and physics". In *Meteorologische Zeitschrift* (2009).

[8] World Meteorological Organization. "Observation components of the Global Observing System". 2020.

[9] Joanne Jeppesen. "ECMWF's use of satellite observations". 2020.

[10] Peter Bauer, Alan Thorpe, and Gilbert Brunet. "The quiet revolution of numerical weather prediction". In *Nature* 525.7567 (2015).

[11] Leonhard Scheck, Martin Weissmann, and Liselotte Bach. "Assimilating visible satellite images for convective-scale numerical weather prediction: A case-study". In *Quarterly Journal of the Royal Meteorological Society* 146.732 (2020).

[12] Lukas Kugler, Jeffrey L. Anderson, and Martin Weissmann. "Potential impact of all-sky assimilation of visible and infrared satellite observations compared with radar reflectivity for convective-scale numerical weather prediction". In *Quarterly Journal of the Royal Meteorological Society* 149.757 (2023).

[13] Alan Turing. "Computing Machinery and Intelligence". In *Mind* LIX.236 (1950).

[14] Ashish Vaswani et al. "Attention is All you Need". In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". 2019.

[16] Tom Brown et al. "Language Models are Few-Shot Learners". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[17] Aaron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". 2016.

[18] Alec Radford et al. "Robust Speech Recognition via Large-Scale Weak Supervision". 2022.

[19] Karan Goel et al. "It's Raw! Audio Generation with State-Space Models". In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[20] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In *IEEE International Conference on Computer Vision*. 2017.

[21] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[22] Robin Rombach et al. “High-Resolution Image Synthesis With Latent Diffusion Models”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[23] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In *Nature* 518.7540 (2015).

[24] David Silver et al. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”. 2017.

[25] Bowen Baker et al. “Emergent Tool Use From Multi-Agent Autocurricula”. In *International Conference on Learning Representations*. 2020.

[26] OpenAI: Marcin Andrychowicz et al. “Learning dexterous in-hand manipulation”. In *The International Journal of Robotics Research* 39.1 (2020).

[27] Jason Wei et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”. In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[28] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In *Nature* 596.7873 (2021).

[29] David Rolnick et al. “Tackling Climate Change with Machine Learning”. In *ACM Comput. Surv.* 55.2 (2022).

[30] Jonas Degrave et al. “Magnetic control of tokamak plasmas through deep reinforcement learning”. In *Nature* 602.7897 (2022).

[31] Amil Merchant et al. “Scaling deep learning for materials discovery”. In *Nature* 624.7990 (2023).

[32] David Pfau et al. “Accurate computation of quantum excited states with neural networks”. In *Science* 385.6711 (2024).

[33] Laura Manduchi et al. “Leveraging Cardiovascular Simulations for In-Vivo Prediction of Cardiac Biomarkers”. 2024.

[34] Suman Ravuri et al. “Skilful precipitation nowcasting using deep generative models of radar”. In *Nature* 597.7878 (2021).

[35] Peter Dueben et al. “Machine learning at ECMWF: A roadmap for the next 10 years”. Tech. rep. ECMWF, 2021.

[36] Julian Mack et al. “Attention-based Convolutional Autoencoders for 3D-Variational Data Assimilation”. In *Computer Methods in Applied Mechanics and Engineering* 372 (2020).

[37] Julien Brajard et al. “Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model”. In *Journal of Computational Science* 44 (2020).

[38] Julien Brajard et al. “Combining data assimilation and machine learning to infer unresolved scale parametrization”. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (2021).

[39] Rossella Arcucci et al. “Deep Data Assimilation: Integrating Deep Learning with Data Assimilation”. In *Applied Sciences* 11.3 (2021).

[40] R. Fablet et al. “Learning Variational Data Assimilation Models and Solvers”. In *Journal of Advances in Modeling Earth Systems* 13.10 (2021).

[41] Thomas Frerix et al. “Variational Data Assimilation with a Learned Inverse Observation Operator”. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[42] Caterina Buizza et al. “Data Learning: Integrating Data Assimilation and Machine Learning”. In *Journal of Computational Science* 58 (2022).

[43] Yu-Hong Yeung, David A. Barajas-Solano, and Alexandre M. Tartakovsky. “Physics-Informed Machine Learning Method for Large-Scale Data Assimilation Problems”. In *Water Resources Research* 58.5 (2022).

[44] Marcin Andrychowicz et al. “Deep Learning for Day Forecasts from Sparse Observations”. 2023.

[45] Remi Lam et al. “Learning skillful medium-range global weather forecasting”. In *Science* 382.6677 (2023).

[46] Simon Lang et al. “AIFS: ECMWF’s data-driven forecasting system”. 2024.

[47] Georg Lentze. “ECMWF’s AI forecasts become operational”. 2025.

[48] Zhuowen Tu. “Learning Generative Models via Discriminative Approaches”. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2007.

[49] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. “Density Ratio Estimation in Machine Learning”. Cambridge University Press, 2012.

[50] Michael U. Gutmann and Aapo Hyvärinen. “Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics”. In *Journal of Machine Learning Research* 13.11 (2012).

[51] Kyle Cranmer, Juan Pavez, and Gilles Louppe. “Approximating Likelihood Ratios with Calibrated Discriminative Classifiers”. 2016.

[52] K Cranmer et al. “Experiments using machine learning to approximate likelihood ratios for mixture models”. In *Journal of Physics: Conference Series* 762.1 (2016).

[53] Aditya Menon and Cheng Soon Ong. “Linking losses for density ratio and class-probability estimation”. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2016.

[54] Markus Stoye et al. “Likelihood-free inference with an improved cross-entropy estimator”. 2018.

[55] Mohamed Ishmael Belghazi et al. “Mutual Information Neural Estimation”. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018.

[56] Conor Durkan, Iain Murray, and George Papamakarios. “On Contrastive Learning for Likelihood-free Inference”. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[57] Benjamin Rhodes, Kai Xu, and Michael U. Gutmann. “Telescoping Density-Ratio Estimation”. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[58] Johann Brehmer et al. “Mining gold from implicit models to improve likelihood-free inference”. In *Proceedings of the National Academy of Sciences* 117.10 (2020).

[59] Joeri Hermans, Volodimir Begy, and Gilles Louppe. “Likelihood-free MCMC with Amortized Approximate Ratio Estimators”. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[60] Benjamin K Miller et al. “Truncated Marginal Neural Ratio Estimation”. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[61] François Rozet. “Arbitrary Marginal Neural Ratio Estimation for Likelihood-free Inference”. PhD thesis. Université de Liège, 2021.

[62] Benjamin K. Miller, Christoph Weniger, and Patrick Forré. "Contrastive Neural Ratio Estimation". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[63] Arnaud Delaunoy et al. "Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[64] Kristy Choi et al. "Density Ratio Estimation via Infinitesimal Classification". In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[65] E. G. Tabak and Cristina V. Turner. "A Family of Nonparametric Density Estimation Algorithms". In *Communications on Pure and Applied Mathematics* 66.2 (2013).

[66] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". 2014.

[67] Danilo Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[68] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". 2016.

[69] George Papamakarios, Theo Pavlakou, and Iain Murray. "Masked Autoregressive Flow for Density Estimation". In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[70] Chin-Wei Huang et al. "Neural Autoregressive Flows". In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018.

[71] Antoine Wehenkel and Gilles Louppe. "Unconstrained Monotonic Neural Networks". In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[72] Conor Durkan et al. "Neural Spline Flows". In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[73] George Papamakarios et al. "Normalizing Flows for Probabilistic Modeling and Inference". In *Journal of Machine Learning Research* 22.57 (2021).

[74] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In *Proceedings of the National Academy of Sciences* 117.48 (2020).

[75] ATLAS Collaboration. "An implementation of neural simulation-based inference for parameter estimation in ATLAS". In *Reports on Progress in Physics* 88.6 (2025).

[76] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In *International Conference on Learning Representations*. 2013.

[77] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In *Foundations and Trends in Machine Learning* 12.4 (2019).

[78] Ian J. Goodfellow et al. "Generative Adversarial Networks". 2014.

[79] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[80] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[81] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[82] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In *International Conference on Learning Representations*. 2021.

[83] Xingchao Liu, Chengyue Gong, and Qiang Liu. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In *International Conference on Learning Representations*. 2023.

[84] Yaron Lipman et al. "Flow Matching for Generative Modeling". In 2023.

[85] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. "Stochastic Interpolants: A Unifying Framework for Flows and Diffusions". 2023.

[86] Isaac Newton. "Philosophiae naturalis principia mathematica". Londini, 1687.

[87] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[88] Chieh-Hsin Lai et al. "The Principles of Diffusion Models". 2025.

[89] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[90] Thomas M. Hamill. "Ensemble-based atmospheric data assimilation". In *Predictability of Weather and Climate*. 2006.

[91] Geir Evensen. "Data Assimilation: The Ensemble Kalman Filter". Springer, 2009.

[92] ECMWF. "IFS documentation CY47R1 - part II: Data assimilation". In *IFS Documentation CY47R1*. ECMWF, 2020.

[93] H. O. Hartley. "Maximum Likelihood Estimation from Incomplete Data". In *Biometrics* (1958).

[94] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In *Journal of the Royal Statistical Society* (1977).

[95] C. F. Jeff Wu. "On the Convergence Properties of the EM Algorithm". In *The Annals of Statistics* (1983).

[96] Geoffrey J McLachlan and Thriyambakam Krishnan. "The EM algorithm and extensions". John Wiley & Sons, 2007.

[97] Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. "Statistical guarantees for the EM algorithm: From population to sample-based analysis". In *The Annals of Statistics* (2017).

[98] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[99] François Rozet and Gilles Louppe. "Score-based Data Assimilation for a Two-Layer Quasi-Geostrophic Model". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2023.

[100] François Rozet et al. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[101] François Rozet et al. "Lost in Latent Space: An Empirical Study of Latent Diffusion Models for Physics Emulation". In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

[102] Gérôme Andry et al. “Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation”. In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2025.

[103] Joeri Hermans et al. “A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful”. In *Transactions on Machine Learning Research* (2022).

[104] Malavika Vasist et al. “Neural posterior estimation for exoplanetary atmospheric retrieval”. In *Astronomy & Astrophysics* 672 (2023).

[105] Ruben Ohana et al. “The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[106] Thomas Savary, François Rozet, and Gilles Louppe. “Training-Free Data Assimilation with GenCast”. In *Tackling Climate Change with Machine Learning Workshop (NeurIPS)*. 2025.

[107] Rudy Morel et al. “Predicting partially observable dynamical systems via diffusion models with a multiscale inference scheme”. In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

[108] Michael McCabe et al. “Walrus: A Cross-Domain Foundation Model for Continuum Dynamics”. 2025.

[109] François Rozet, Arnaud Delaunoy, Benjamin Miller, et al. “LAMPE: Likelihood-free amortized posterior estimation”. 2021.

[110] François Rozet et al. “Zuko: Normalizing flows in PyTorch”. 2022.

[111] François Rozet. “Azula: Diffusion models in PyTorch”. 2024.

II
Matter

2 Score-based Data Assimilation

> *A simple chain is an infinite sequence $x_1, x_2, \ldots, x_k, x_{k+1}, \ldots$ of variables connected in such a way that x_{k+1} for any k is independent of $x_1, x_2, \ldots, x_{k-1}$ in case x_k is known.*
>
> — Andrei Markov (1906)

Addendum

This chapter has previously been published as

> François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

As the leading author, François came up with the method, formalized it, conducted the experiments, interpreted the results, and wrote the manuscript. Gilles supervised the project, suggested experiments, and participated in the writing and literature review.

For the version presented in this chapter, we have slightly edited the mathematical notations from the original publication. Notably, the diffusion time notation $x(t)$ has been replaced by the leaner x_t, which is more common in nowadays literature and more consistent with the rest of this document. The text itself remains unchanged, although we have enhanced it with a few retrospective comments.

Abstract

Data assimilation, in its most comprehensive form, addresses the Bayesian inverse problem of identifying plausible state trajectories that explain noisy or incomplete observations of stochastic dynamical systems. Various approaches have been proposed to solve this problem, including particle-based and variational methods. However, most algorithms depend on the transition dynamics for inference, which becomes intractable for long time horizons or for high-dimensional systems with complex dynamics, such as oceans or atmospheres. In this work, we introduce score-based data assimilation for trajectory inference. We learn a score-based generative model of state trajectories based on the key insight that the score of an arbitrarily long trajectory can be decomposed into a series of scores over short segments. After training, inference is carried out using the score model, in a non-autoregressive manner by generating all states simultaneously. Quite distinctively, we decouple the observation model from the training procedure and use it only at inference to guide the generative process, which enables a wide range of zero-shot observation scenarios. We present theoretical and empirical evidence supporting the effectiveness of our method.

2.1 Introduction

Data assimilation (DA) [3–11] is at the core of many scientific domains concerned with the study of complex dynamical systems such as atmospheres, oceans or climates. The purpose of DA is to infer the state of a system evolving over time based on various sources of imperfect information, including sparse, intermittent, and noisy observations.

Formally, let $x^{1:L} = (x^1, x^2, \dots, x^L) \in \mathbb{R}^{L \times D}$ denote a trajectory of states in a discrete-time stochastic dynamical system and $p(x^{i+1} \mid x^i)$ be the transition dynamics from state x^i to state x^{i+1}. An observation $y \in \mathbb{R}^M$ of the state trajectory $x^{1:L}$ follows an observation process $p(y \mid x^{1:L})$, generally formulated as $y = \mathcal{A}(x^{1:L}) + \eta$, where the measurement function $\mathcal{A} : \mathbb{R}^{L \times D} \mapsto \mathbb{R}^M$ is often non-linear and the observational error $\eta \in \mathbb{R}^M$ is a stochastic additive term that accounts for instrumental noise and systematic uncertainties. In this framework, the goal of DA is to solve the inverse problem of inferring plausible trajectories $x^{1:L}$ given an observation y, that is, to estimate the trajectory posterior

$$p(x^{1:L} \mid y) = \frac{p(y \mid x^{1:L})}{p(y)} \, p(x^1) \prod_{i=1}^{L-1} p(x^{i+1} \mid x^i) \tag{2.1}$$

where the initial state prior $p(x^1)$ is commonly referred to as background [7–11]. In geosciences, the amount of data available is generally insufficient to recover the full state of the system from the observation alone [10]. For this reason, the physical model underlying the transition dynamics is of paramount importance to fill in spatial and temporal gaps in the observation.

State-of-the-art approaches to data assimilation are based on variational assimilation [3, 4, 7–9]. Many of these approaches formulate the task as a maximum-a-posteriori (MAP) estimation problem and solve it by maximizing the log-posterior density $\log p(x^{1:L} \mid y)$ via gradient ascent. Although this approach only produces a point estimate of the trajectory posterior, its cost can already be substantial for problems of the size and complexity of geophysical systems, since it requires differentiating through the physical model. The amount of data that can be assimilated is therefore restricted because of computational limitations. For example, only a small volume of the available satellite data is exploited for operational forecasts and yet, even with these restrictions, data assimilation accounts for a significant fraction of the computational cost for modern numerical weather prediction [12, 13]. Recent work has shown that deep learning can be used in a variety of ways to improve the computational efficiency of data assimilation, increase the reconstruction performance by estimating unresolved scales after data assimilation, or integrate multiple sources of observations [14–21].

Contributions In this work, we propose a novel approach to data assimilation based on score-based generative models. Leveraging the Markovian structure of dynamical systems, we train a score network from short segments of trajectories which is then capable of generating physically consistent and arbitrarily-long state trajectories. The observation model is decoupled from the score network and used only during assimilation to guide the generative process, which allows for a wide range of zero-shot observation scenarios. Our approach provides an accurate approximation of the whole trajectory posterior – it is not limited to point estimates – without simulating or differentiating through the physical model. The code for all experiments is made available at `https://github.com/francois-rozet/sda`.

2.2 Background

Score-based generative models have recently shown remarkable capabilities, powering many of the latest advances in image, video or audio generation [22–29]. In this section, we review score-based generative models and outline how they can be used for solving inverse problems.

Continuous-time score-based generative models Adapting the formulation of Song et al. [30], samples $x \in \mathbb{R}^D$ from a distribution $p(x)$ are progressively perturbed through a continuous-time diffusion process expressed as a linear stochastic differential equation (SDE)

$$\mathrm{d}x_t = f_t\, x_t\, \mathrm{d}t + g_t\, \mathrm{d}w_t \tag{2.2}$$

where $f_t \in \mathbb{R}$ is the drift coefficient, $g_t \in \mathbb{R}$ is the diffusion coefficient, $w_t \in \mathbb{R}^D$ denotes a Wiener process (standard Brownian motion) and $x_t \in \mathbb{R}^D$ is the perturbed sample at time $t \in [0, 1]$. Because the SDE is linear with respect to x_t, the perturbation kernel from x to x_t is Gaussian and takes the form

$$p(x_t \mid x) = \mathcal{N}(x_t \mid \alpha_t\, x, \Sigma_t) \tag{2.3}$$

where α_t and $\Sigma_t = \sigma_t^2 I$ can be derived analytically from f_t and g_t [31, 32]. Denoting $p(x_t)$ the marginal distribution of x_t, we impose that $\alpha_0 = 1$ and $\sigma_0 \ll 1$, such that $p(x_0) \approx p(x)$, and we chose the coefficients f_t and g_t such that the influence of the initial sample x on the final perturbed sample x_1 is negligible with respect to the noise level – that is, $p(x_1) \approx \mathcal{N}(0, \Sigma_1)$. The variance exploding (VE) and variance preserving (VP) SDEs [30, 33, 34] are widespread examples satisfying these constraints.

Crucially, the time reversal of the forward SDE (2.2) is given by a reverse SDE [30, 35]

$$\mathrm{d}x_t = \left[f_t\, x_t - g_t^2\, \nabla_{x_t} \log p(x_t)\right] \mathrm{d}t + g_t\, \mathrm{d}w_t. \tag{2.4}$$

That is, we can draw noise samples $x_1 \sim \mathcal{N}(0, \Sigma_1)$ and gradually remove the noise therein to obtain data samples $x_0 \sim p(x_0)$ by simulating the reverse SDE from $t = 1$ to 0. This requires access to the quantity $\nabla_{x_t} \log p(x_t)$ known as the score of $p(x_t)$.

Denoising score matching In practice, the score $\nabla_{x_t} \log p(x_t)$ is approximated by a neural network $s_\phi(x_t, t)$, named the score network, which is trained to solve the denoising score matching objective [30, 36, 37]

$$\arg\min_\phi \mathbb{E}_{p(x)p_t p(x_t|x)} \left[\sigma_t^2 \left\| s_\phi(x_t, t) - \nabla_{x_t} \log p(x_t \mid x)\right\|_2^2\right] \tag{2.5}$$

where $p_t = \mathcal{U}(0, 1)$. The theory of denoising score matching ensures that $s_\phi(x_t, t) \approx \nabla_{x_t} \log p(x_t)$ for a sufficiently expressive score network. After training, the score network is plugged into the reverse SDE (2.4), which is then simulated using an appropriate discretization scheme [30, 32, 38, 39].

In practice, the high variance of $\nabla_{x_t} \log p(x_t \mid x)$ near $t = 0$ makes the optimization of (2.5) unstable [32]. To mitigate this issue, a slightly different parameterization $\epsilon_\phi(x_t, t) = -\sigma_t\, s_\phi(x_t, t)$ of the score network is often used, which leads to the otherwise equivalent objective [32, 34, 38]

We later found out that the EDM [40] or flow [41] parameterizations would have been better choices.

$$\arg\min_\phi \mathbb{E}_{p(x)p_t p(\epsilon)} \left[\left\| \epsilon_\phi(\alpha_t\, x + \sigma_t\, \epsilon, t) - \epsilon\right\|_2^2\right] \tag{2.6}$$

where $p(\epsilon) = \mathcal{N}(0, I)$. In the following, we keep the score network notation $s_\phi(x_t, t)$ for convenience, even though we adopt the parameterization $\epsilon_\phi(x_t, t)$ and its objective for our experiments.

Zero-shot inverse problems With score-based generative models, we can generate samples from the unconditional distribution $p(x_0) \approx p(x)$. To solve inverse problems, however, we need to sample from the posterior distribution $p(x \mid y)$. This could be accomplished by training a conditional score network $s_\phi(x_t, t \mid y)$ to approximate the posterior score $\nabla_{x_t} \log p(x_t \mid y)$ and plugging it into the reverse SDE (2.4). However, this would require data pairs (x, y) during training and one would need to retrain a new score network each time the observation process $p(y \mid x)$ changes. Instead, many have observed [30, 42–45] that the posterior score can be decomposed into two terms thanks to Bayes' rule

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y \mid x_t). \tag{2.7}$$

Since the prior score $\nabla_{x_t} \log p(x_t)$ can be approximated with a single score network, the remaining task is to estimate the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$. Assuming a differentiable measurement function $\mathcal{A}$ and a Gaussian observation process $p(y \mid x) = \mathcal{N}(y \mid \mathcal{A}(x), \Sigma_y)$, Chung et al. [45] propose the approximation

$$p(y \mid x_t) = \int p(y \mid x)\, p(x \mid x_t)\, \mathrm{d}x \approx \mathcal{N}\left(y \mid \mathcal{A}(\hat{x}(x_t)), \Sigma_y\right) \tag{2.8}$$

where the mean $\hat{x}(x_t) = \mathbb{E}_{p(x|x_t)}[x]$ is given by Tweedie's formula [46, 47]

$$\mathbb{E}_{p(x|x_t)}[x] = \frac{x_t + \sigma_t^2 \nabla_{x_t} \log p(x_t)}{\alpha_t} \tag{2.9}$$

$$\approx \frac{x_t + \sigma_t^2\, s_\phi(x_t, t)}{\alpha_t}. \tag{2.10}$$

As the log-likelihood of a multivariate Gaussian is known analytically and $s_\phi(x_t, t)$ is differentiable, we can compute the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ with this approximation in zero-shot, that is, without training any other network than $s_\phi(x_t, t)$.

2.3 Score-based data assimilation

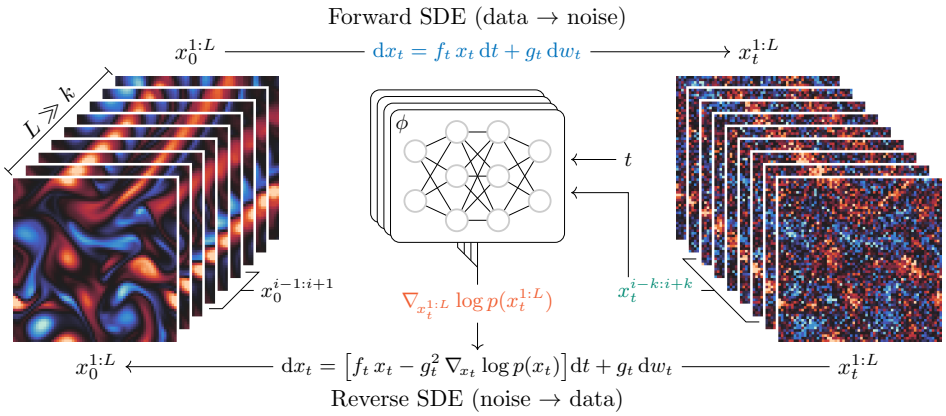
Coming back to our initial inference problem, we want to approximate the trajectory posterior $p(x^{1:L} \mid y)$ of a dynamical system. To do so with score-based generative modeling, we need to estimate the posterior score $\nabla_{x_t^{1:L}} \log p(x_t^{1:L} \mid y)$, which we choose to decompose into prior and likelihood terms, as in (2.7), to enable a wide range of zero-shot observation scenarios.

In typical data assimilation settings, the high-dimensionality of each state x^i (e.g. the state of atmospheres or oceans) combined with potentially long trajectories would require an impractically large score network $s_\phi(x_t^{1:L}, t)$ to estimate the prior score $\nabla_{x_t^{1:L}} \log p(x_t^{1:L})$ and a proportional amount of data for training, which could be prohibitive if data is scarce or if the physical model is expensive to simulate. To overcome this challenge, we leverage the Markovian structure of dynamical systems to approximate the prior score with a series of local scores, which are easier to learn, as explained in Section 2.3.1. In Section 2.3.2, we build upon diffusion posterior sampling (DPS) [45] to propose a new approximation for the likelihood score $\nabla_{x_t^{1:L}} \log p(y \mid x_t^{1:L})$, which we find more appropriate for posterior inference. Finally, in Section 2.3.3, we describe our sampling procedure inspired from predictor-corrector sampling [30]. Our main contribution, named score-based data assimilation (SDA), is the combination of these three components.

This section's title is a reference to OSS 117, a classic french comedy movie. To our delight, one of the reviewers noticed this reference, and replied with another reference: "The blanket is good."

2.3.1 How is your blanket?

Given a set of random variables $x^{1:L} = \{x^1, x^2, \dots, x^L\}$, it is sometimes possible to find a small Markov blanket $x^{b_i} \subseteq x^{\neq i}$ such that $p(x^i \mid x^{\neq i}) = p(x^i \mid x^{b_i})$ for each



The lines connecting the stack of networks and the score $\nabla_{x_t^{1:L}} \log p(x_t^{1:L})$ represent Algorithm 2.

Figure 2.1. Trajectories $x^{1:L}$ of a dynamical system are transformed to noise via a diffusion process. Reversing this process generates new trajectories, but requires the score of $p(x_t^{1:L})$. We approximate it by combining the outputs of a score network over sub-segments of $x_t^{1:L}$.

element x^i using our knowledge of the set's structure. It follows that each element $\nabla_{x^i} \log p(x^{1:L})$ of the full score $\nabla_{x^{1:L}} \log p(x^{1:L})$ can be determined locally, that is, only using its blanket;

$$\nabla_{x^i} \log p(x^{1:L}) = \nabla_{x^i} \log p(x^i \mid x^{\neq i}) + \nabla_{x^i} \log p(x^{\neq i}) \tag{2.11}$$

$$= \nabla_{x^i} \log p(x^i \mid x^{b_i}) + \nabla_{x^i} \log p(x^{b_i}) = \nabla_{x^i} \log p(x^i, x^{b_i}). \tag{2.12}$$

This property generally does not hold for the diffusion-perturbed set $x_t^{1:L}$ as there is no guarantee that $x_t^{b_i}$ is a Markov blanket of the element x_t^i. However, there exists a set of indices $\bar{b}_i \supseteq b_i$ such that

$$\nabla_{x_t^i} \log p(x_t^{1:L}) \approx \nabla_{x_t^i} \log p(x_t^i, x_t^{\bar{b}_i}) \tag{2.13}$$

is a good approximation for all $t \in [0, 1]$. That is, $x_t^{\bar{b}_i}$ is a "pseudo" Markov blanket of x_t^i. In the worst case, $\bar{b}_i$ contains all indices except i, but we argue that, for some structures, there is a set $\bar{b}_i$ not much larger than b_i that satisfies (2.13). Our rationale is that, since we impose the initial noise to be negligible, we know that $x_t^{b_i}$ becomes indistinguishable from x^{b_i} as t approaches 0. Furthermore, as t grows and noise accumulates, the mutual information between elements x_t^i and x_t^j decreases to finally reach 0 when $t = 1$. Hence, even if $\bar{b}_i = b_i$, the pseudo-blanket approximation (2.13) already holds near $t = 0$ and $t = 1$. In between, even though the approximation remains unbiased (see Appendix 2.A), the structure of the set becomes decisive. If it is known and present enough regularities/symmetries, (2.13) could and should be exploited within the architecture of the score network $s_\phi(x_t^{1:L}, t)$.

In the case of dynamical systems, the set $x^{1:L}$ is by definition a first-order Markov chain and the minimal Markov blanket of an element x^i is $x^{b_i} = \{x^{i-1}, x^{i+1}\}$. For the perturbed element x_t^i, the pseudo-blanket $x_t^{\bar{b}_i}$ can take the form of a window surrounding x_t^i, that is $\bar{b}_i = \{i - k, \dots, i + k\} \setminus \{i\}$ with $k \geq 1$. The value of k is dependent on the problem, but we argue, supported by our experiments, that it is generally much smaller than the chain's length L. Hence, a fully convolutional neural network (FCNN) with a narrow receptive field is well suited to the task, and long-range capabilities would be wasted resources. Importantly, if the receptive field is $2k + 1$, the network can be trained on segments $x^{i-k:i+k}$ instead of the full chain $x^{1:L}$, thereby drastically reducing training costs. More generally, we can train a local score network (see Algorithm 1)

In hindsight, systems governed by slowly evolving dynamics require much wider windows than we expected.

$$s_\phi(x_t^{i-k:i+k}, t) \approx \nabla_{x_t^{i-k:i+k}} \log p(x_t^{i-k:i+k}) \tag{2.14}$$

Algorithm 1 Training $\epsilon_\phi(x_t^{i-k:i+k}, t)$

for $i = 1$ to N **do**
  $x^{1:L} \sim p(x^{1:L})$
  $i \sim \mathcal{U}(\{k+1, \dots, L-k\})$
  $t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0, I)$
  $x_t^{i-k:i+k} \leftarrow \alpha_t\, x^{i-k:i+k} + \sigma_t\, \epsilon$
  $\ell \leftarrow \|\epsilon_\phi(x_t^{i-k:i+k}, t) - \epsilon\|_2^2$
  $\phi \leftarrow \textsc{GradientDescent}(\phi, \nabla_\phi\, \ell)$

Algorithm 2 Composing $s_\phi(x_t^{i-k:i+k}, t)$

function $s_\phi(x_t^{1:L}, t)$
  $s_{1:k+1} \leftarrow s_\phi(x_t^{1:2k+1}, t)[:k+1]$
  for $i = k+2$ to $L-k-1$ **do**
    $s_i \leftarrow s_\phi(x_t^{i-k:i+k}, t)[k+1]$
  $s_{L-k:L} \leftarrow s_\phi(x_t^{L-2k:L}, t)[k+1:]$
  return $s^{1:L}$

such that its $k+1$-th element approximates the score of the i-th state $\nabla_{x_t^i} \log p(x_t^{1:L})$. We also have that the k first elements of $s_\phi(x_t^{1:2k+1}, t)$ approximate the score of the k first states $\nabla_{x_t^{1:k}} \log p(x_t^{1:L})$ and the k last elements of $s_\phi(x_t^{L-2k:L}, t)$ approximate the score of the k last states $\nabla_{x_t^{L-k:L}} \log p(x_t^{1:L})$. Hence, we can apply the local score network on all sub-segments $x_t^{i-k:i+k}$ of $x_t^{1:L}$, similar to a convolution kernel, and combine the outputs (see Algorithm 2) to get an approximation of the full score $\nabla_{x_t^{1:L}} \log p(x_t^{1:L})$. Note that we can either condition the score network with i or assume the statistical stationarity of the chain, that is $p(x^i) = p(x^{i+1})$.

2.3.2 Stable likelihood score

Due to approximation and numerical errors in $\hat{x}(x_t)$, computing the score $\nabla_{x_t} \log p(y \mid x_t)$ with the likelihood approximation (2.8) is very unstable, especially in the low signal-to-noise regime, that is when $\sigma_t \gg \alpha_t$. This incites Chung et al. [45] to replace the covariance Σ_y by the identity I and rescale the likelihood score with respect to $\|y - \mathcal{A}(\hat{x}(x_t))\|$ to stabilize the sampling process. These modifications introduce a significant error in the approximation as they greatly affect the norm of the likelihood score.

We argue that the instability is due to (2.8) being only exact if the variance of $p(x \mid x_t)$ is null or negligible, which is not the case when $t > 0$. Instead, Adam et al. [44] and Meng et al. [49] approximate the covariance of $p(x \mid x_t)$ with ${}^{\Sigma_t}/_{\alpha_t^2}$, which is valid as long as the prior $p(x)$ is Gaussian with a large diagonal covariance Σ_x. We motivate in Appendix 2.B the more general covariance approximation ${}^{\sigma_t^2}/_{\alpha_t^2}\Gamma$, where the matrix Γ depends on the eigendecomposition of Σ_x. Then, taking inspiration from the extended Kalman filter, we approximate the perturbed likelihood as

This was the start of our journey towards better likelihood score approximations, which eventually led to the development of MMPS [48].

$$p(y \mid x_t) \approx \mathcal{N}\left(y \mid \mathcal{A}(\hat{x}(x_t)), \Sigma_y + \frac{\sigma_t^2}{\alpha_t^2} A \Gamma A^T\right) \tag{2.15}$$

where $A = \partial_x \mathcal{A}\,|_{\hat{x}(x_t)}$ is the Jacobian of $\mathcal{A}$. In practice, to simplify the approximation, the term $A\Gamma A^T$ can often be replaced by a constant (diagonal) matrix. We find that computing the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ with this new approximation (see Algorithm 3) is stable enough that rescaling it or ignoring Σ_y is unnecessary.

2.3.3 Predictor-Corrector sampling

To simulate the reverse SDE, we adopt the exponential integrator (EI) discretization scheme introduced by Zhang et al. [32]

$$x_{t-\Delta t} \leftarrow \frac{\alpha_{t-\Delta t}}{\alpha_t} x_t + \left(\frac{\alpha_{t-\Delta t}}{\alpha_t} - \frac{\sigma_{t-\Delta t}}{\sigma_t}\right) \sigma_t^2\, s_\phi(x_t, t) \tag{2.16}$$

which coincides with the deterministic DDIM [38] sampling algorithm when the variance preserving SDE [34] is used. However, as we approximate both the prior and

likelihood scores, errors accumulate along the simulation and cause it to diverge, leading to low-quality samples. To prevent errors from accumulating, we perform (see Algorithm 4) a few steps of Langevin Monte Carlo (LMC) [50, 51]

$$x_t \leftarrow x_t + \delta\, s_\phi(x_t, t) + \sqrt{2\delta}\,\epsilon \tag{2.17}$$

where $\epsilon \sim \mathcal{N}(0, I)$, between each step of the discretized reverse SDE (2.16). In the limit of an infinite number of LMC steps with a sufficiently small step size $\delta \in \mathbb{R}_+$, simulated samples are guaranteed to follow the distribution implicitly defined by our approximation of the posterior score at each time t, meaning that the errors introduced by the pseudo-blanket (2.13) and likelihood (2.15) approximations do not accumulate. In practice, we find that few LMC steps are necessary. Song et al. [30] introduced a similar strategy, named predictor-corrector (PC) sampling, to correct the errors introduced by the discretization of the reverse SDE.

2.4 Results

We demonstrate the effectiveness of score-based data assimilation on two chaotic dynamical systems: the Lorenz 1963 [52] and Kolmogorov flow [53] systems. The former is a simplified mathematical model for atmospheric convection. Its low dimensionality enables posterior inference using classical sequential Monte Carlo methods [54, 55] such as the bootstrap particle filter [56]. This allows us to compare objectively our posterior approximations against the ground-truth posterior. The second system considers the state of a two-dimensional turbulent fluid subject to Kolmogorov forcing [53]. The evolution of the fluid is modeled by the Navier-Stokes equations, the same equations that underlie the models of oceans and atmospheres. This task provides a good understanding of how SDA would perform in typical data assimilation applications, although our analysis is primarily qualitative due to the unavailability of reliable assessment tools for systems of this scale.

For both systems, we employ as diffusion process the variance preserving SDE with a cosine schedule [57], that is $\alpha_t = \cos(\omega t)^2$ with $\omega = \arccos\sqrt{10^{-3}}$ and $\sigma_t = \sqrt{1-\alpha_t^2}$. The score networks are trained once and then evaluated under various observation scenarios. Unless specified otherwise, we estimate the posterior score according to Algorithm 3 with $\Gamma = 10^{-2} I$ and simulate the reverse SDE (2.4) according to Algorithm 4 in 256 evenly spaced discretization steps.

2.4.1 Lorenz 1963

The state $x = (a, b, c) \in \mathbb{R}^3$ of the Lorenz system evolves according to a system of ordinary differential equations

$$\begin{aligned} \dot{a} &= \sigma(b - a) \\ \dot{b} &= a(\rho - c) - b \\ \dot{c} &= ab - \beta c \end{aligned} \tag{2.18}$$

where $\sigma = 10$, $\rho = 28$ and $\beta = \frac{8}{3}$ are parameters for which the system exhibits a chaotic behavior. We denote $\tilde{a}$ and $\tilde{c}$ the standardized (zero mean and unit variance) versions of a and c, respectively. As our approach assumes a discrete-time stochastic dynamical system, we consider a transition process of the form $x^{i+1} = \mathcal{M}(x^i) + \eta$, where $\mathcal{M} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is the integration of the differential equations (2.18) for $\Delta = 0.025$ time units and $\eta \sim \mathcal{N}(0, \Delta I)$ represents Brownian noise.

We generate 1024 independent trajectories of 1024 states, which are split into training (80 %), validation (10 %) and evaluation (10 %) sets. The initial states are drawn from the

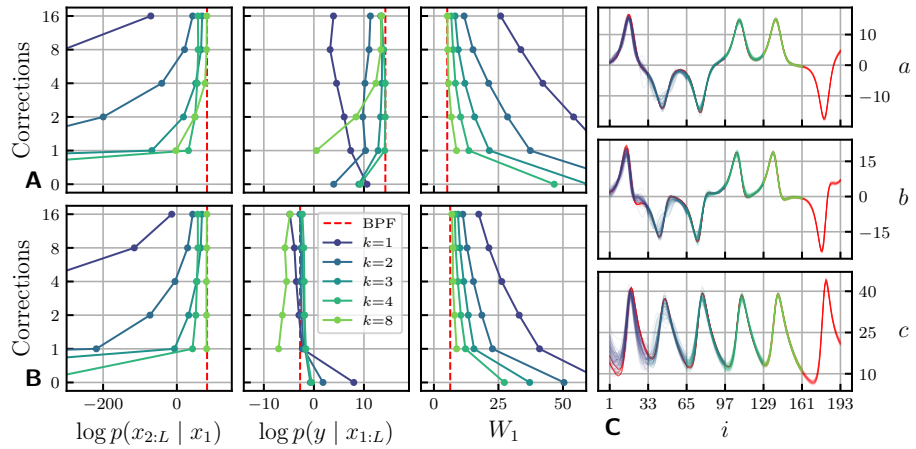


Figure 2.2. Average posterior summary statistics over 64 observations from the low (**A**) and high (**B**) frequency observation processes. We observe that, as k and the number of corrections C increase, the statistics of the approximate posteriors get closer to the ground-truth, in red, which means they are getting more accurate. However, increasing k and C improves the quality of posteriors with decreasing return, such that all posteriors with $k \geq 3$ and $C \geq 2$ are almost equivalent. This is visible in **C**, where we display trajectories inferred ($C = 2$) for an observation of the low frequency observation process. For readability, we allocate a segment of 32 states to each k instead of overlapping all 192 states. Note that the Wasserstein distance between the ground-truth posterior and itself is not zero as it is estimated with a finite number (1024) of samples.

statistically stationary regime of the system. We consider two score network architectures: fully-connected local score networks for small k ($k \leq 4$) and fully-convolutional score networks for large k. Architecture and training details for each k are provided in Appendix 2.D.

We first study the impact of k (see Section 2.3.1) and the number of LMC corrections (see Section 2.3.3) on the quality of the inferred posterior. We consider two simple observation processes $\mathcal{N}(y \mid \tilde{a}^{1:L:8}, 0.05^2 I)$ and $\mathcal{N}(y \mid \tilde{a}^{1:L}, 0.25^2 I)$. The former observes the state at low frequency (every eighth step) with low noise, while the latter observes the state at high frequency (every step) with high noise. For both processes, we generate an observation y for a trajectory of the evaluation set (truncated at $L = 65$) and apply the bootstrap particle filter (BPF) to draw 1024 trajectories $x^{1:L}$ from the ground-truth posterior $p(x^{1:L} \mid y)$. We use a large number of particles (2^{16}) to ensure convergence. Then, using SDA, we sample 1024 trajectories from the approximate posterior $q(x^{1:L} \mid y)$ defined by each score network. We compare the approximate and ground-truth posteriors with three summary statistics: the expected log-prior $\mathbb{E}_{q(x^{1:L}|y)}[\log p(x^{2:L} \mid x^1)]$, the expected log-likelihood $\mathbb{E}_{q(x^{1:L}|y)}[\log p(y \mid x^{1:L})]$ and the Wasserstein distance $W_1(p, q)$ in trajectory space. We repeat the procedure for 64 observations and different number of corrections ($\tau = 0.25$, see Algorithm 4) and present the results in Figure 2.2. To paraphrase, SDA is able to reproduce the ground-truth posterior accurately. Interestingly, accuracy can be traded off for computational efficiency: fewer corrections leads to faster inference at the potential expense of physical consistency.

Another advantage of SDA over variational data assimilation approaches is that it targets the whole posterior distribution instead of point estimates, which allows to identify when several scenarios are plausible. As a demonstration, we generate an observation from the observation process $p(y \mid x^{1:L}) = \mathcal{N}(y \mid \tilde{c}^{1:L:4}, 0.1^2 I)$ and infer plausible trajectories with SDA ($k = 4$, $C = 2$). Several modes are identified in the posterior, which we illustrate in Figure 2.3.

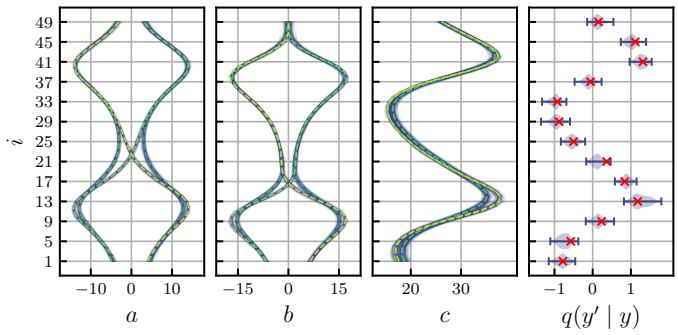


Figure 2.3. Example of multi-modal posterior inference with SDA. We identify four modes (dashed lines) in the inferred posterior. All modes are consistent with the observation (red crosses), as demonstrated by the posterior predictive distribution $q(y' \mid y) = \mathbb{E}_{q(x^{1:L}|y)}[p(y' \mid x^{1:L})]$.

2.4.2 Kolmogorov flow

Incompressible fluid dynamics are governed by the Navier-Stokes equations

$$\begin{aligned} \dot{\boldsymbol{u}} &= -\boldsymbol{u}\nabla\boldsymbol{u} + \frac{1}{Re}\nabla^2\boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f} \\ 0 &= \nabla \cdot \boldsymbol{u} \end{aligned} \tag{2.19}$$

where $\boldsymbol{u}$ is the velocity field, Re is the Reynolds number, ρ is the fluid density, p is the pressure field and $\boldsymbol{f}$ is the external forcing. Following Kochkov et al. [58], we choose a two-dimensional domain $[0, 2\pi]^2$ with periodic boundary conditions, a large Reynolds number $Re = 10^3$, a constant density $\rho = 1$ and an external forcing $\boldsymbol{f}$ corresponding to Kolmogorov forcing with linear damping [53, 59]. We use the `jax-cfd` library [58] to solve the Navier-Stokes equations (2.19) on a 256×256 domain grid. The states x^i are snapshots of the velocity field $\boldsymbol{u}$, coarsened to a 64×64 resolution, and the integration time between two such snapshots is $\Delta = 0.2$ time units. This corresponds to 82 integration

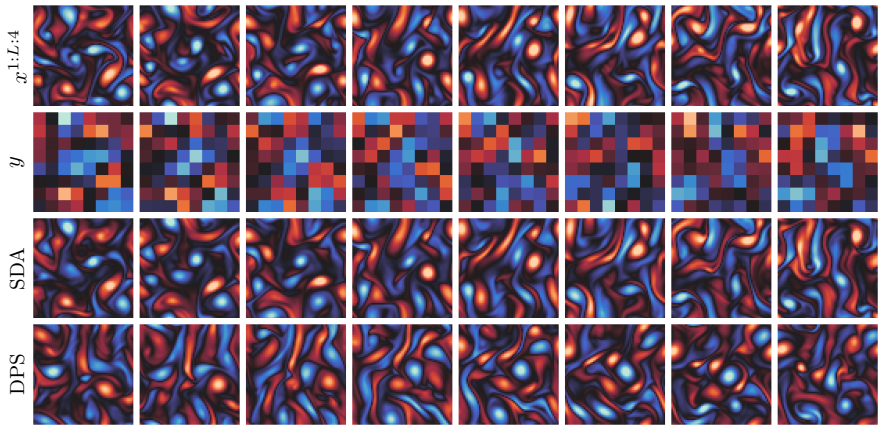


Figure 2.4. Example of sampled trajectory from coarse, intermittent and noisy observations. States are visualized by their vorticity field $\omega = \nabla \times \boldsymbol{u}$, that is the curl of the velocity field. Positive values (red) indicate clockwise rotation and negative values (blue) indicate counter-clockwise rotation. SDA closely recovers the original trajectory, despite the limited amount of available data. Replacing SDA's likelihood score approximation with the one of DPS [45] yields trajectories inconsistent with the observation.

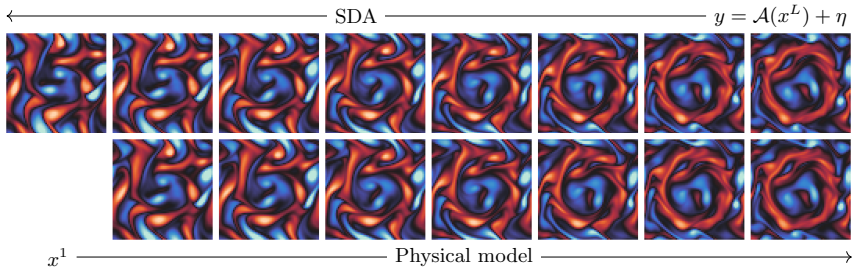


Figure 2.5. A trajectory consistent with an unlikely observation of the final state x^L is generated with SDA. To verify whether the trajectory is realistic and not hallucinated, we plug its initial state x^1 into the physical model and obtain an almost identical trajectory. This indicates that SDA is not simply interpolating between observations, but rather propagates information in a manner consistent with the physical model, even in unlikely scenarios.

To this day, this figure remains one of our favorites. We vividly remember our excitement as we discovered the result of this silly experiment.

steps of the forward Euler method, which would be expensive to differentiate through repeatedly, as required by gradient-based data assimilation approaches.

We generate 1024 independent trajectories of 64 states, which are split into training (80 %), validation (10 %) and evaluation (10 %) sets. The initial states are drawn from the statistically stationary regime of the system. We consider a local score network with $k = 2$. As states take the form of 64×64 images with two velocity channels, we use a U-Net [60] inspired network architecture. Architecture and training details are provided in Appendix 2.D.

We first apply SDA to a classic data assimilation problem. We take a trajectory of length $L = 32$ from the evaluation set and observe the velocity field every four steps, coarsened to a resolution 8×8 and perturbed by a moderate Gaussian noise ($\Sigma_y = 0.1^2 I$). Given the observation, we sample a trajectory with SDA ($C = 1$, $\tau = 0.5$) and find that it closely recovers the original trajectory, as illustrated in Figure 2.4. A similar experiment where we modify the amount of spatial information is presented in Figure 2.8. When data is insufficient to identify the original trajectory, SDA extrapolates a physically plausible scenario while remaining consistent with the observation, which can also be observed in Figure 2.6 and 2.7.

Finally, we investigate whether SDA generalizes to unlikely scenarios. We design an observation process that probes the vorticity of the final state x^L in a circle-shaped sub-domain. Then, we sample a trajectory ($C = 1$, $\tau = 0.5$) consistent with a uniform positive vorticity observation in this sub-domain, which is unlikely, but not impossible. The result is discussed in Figure 2.5.

2.5 Conclusion

Impact In addition to its contributions to the field of data assimilation, this work presents new technical contributions to the field of score-based generative modeling.

First, we provide new insights on how to exploit conditional independencies (Markov blankets) in sets of random variables to build and train score-based generative models. Based on these findings, we are able to generate/infer simultaneously all the states of arbitrarily long Markov chains $x^{1:L}$, while only training score models on short segments $x^{i-k:i+k}$, thereby reducing the training costs and the amounts of training data required. The decomposition of the global score into local scores additionally allows for better

parallelization at inference, which could be significant depending on available hardware. Importantly, the pseudo-blanket approximation (2.13) is not limited to Markov chains, but could be applied to any set of variables $x^{1:L}$, as long as some structure is known.

Second, we motivate and introduce a novel approximation (2.15) for the perturbed likelihood $p(y \mid x_t)$, when the likelihood $p(y \mid x)$ is assumed (linear or non-linear) Gaussian. We find that computing the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ with this new approximation leads to accurate posterior inference, without the need for stability tricks [45]. This contribution can be trivially adapted to many tasks such as inpainting, deblurring, super-resolution or inverse problems in scientific fields [43–45].

Limitations From a computational perspective, even though SDA does not require simulating or differentiating through the physical model, inference remains limited by the speed of the simulation of the reverse SDE. Accelerating sampling in score-based generative models is an active area of research [32, 38, 39, 61] with promising results which would be worth exploring in the context of our method.

Regarding the quality of our results, we empirically demonstrate that SDA provides accurate approximations of the whole posterior, especially as k and the number of LMC corrections C increase. However, our approximations (2.13) and (2.15) introduce a certain degree of error, which precise impact on the resulting posterior remains to be theoretically quantified. Furthermore, although the Kolmogorov system is high-dimensional (tens of thousands of dimensions) with respect to what is approachable with classical posterior inference methods, it remains small in comparison to the millions of dimensions of some operational DA systems. Whether SDA would scale well to such applications is an open question and will present serious engineering challenges.

Another limitation of our work is the assumption that the dynamics of the system are shared by all trajectories. In particular, if a parametric physical model is used, all trajectories are assumed to share the same parameters. For this reason, SDA is not applicable to settings where fitting the model parameters is also required, or at least not without further developments. Some approaches [62–65] tackle this task, but they remain limited to low-dimensional settings. Additionally, if a physical model is used to generate synthetic training data, instead of relying on real data, one can only expect SDA to be as accurate as the model itself. This is a limitation shared by any model-based approach and robust assimilation under model misspecification or distribution shift is left as an avenue for future research.

Finally, posterior inference over entire state trajectories is not always necessary. In forecasting tasks, inferring the current state of the dynamical system is sufficient and likely much less expensive. In this setting, data assimilation reduces to a state estimation problem for which classical methods such as the Kalman filter [66] or its nonlinear extensions [67, 68] provide strong baselines. Many deep learning approaches have also been proposed to bypass the physical model entirely and learn instead a generative model of plausible forecasts from past observations only [69–72].

Related work A number of previous studies have investigated the use of deep learning to improve the quality and efficiency of data assimilation. Mack et al. [14] use convolutional auto-encoders to project the variational data assimilation problem into a lower-dimensional space, which simplifies the optimization problem greatly. Frerix et al. [16] use a deep neural network to predict the initial state of a trajectory given the observations. This prediction is then used as a starting point for traditional (4D-Var) variational data assimilation methods, which proves to be more effective than starting at random. This strategy is also possible with SDA (using a trajectory sampled with SDA as a starting point) and could help cover multiple modes of the posterior distribution.

Finally, Brajard et al. [19] address the problem of simultaneously learning the transition dynamics and estimating the trajectory, when only the observation process is known.

Beyond data assimilation, SDA closely relates to the broader category of sequence models, which have been studied extensively for various types of data, including text, audio, and video. The latest advances demonstrate that score-based generative models achieve remarkable results on the most demanding tasks. Kong et al. [28] and Goel et al. [29] use score-based models to generate long audio sequences non-autoregressively. Ho et al. [25] train a score-based generative model for a fixed number of video frames and use it autoregressively to generate videos of arbitrary lengths. Conversely, our approach is non-autoregressive which allows to generate and condition all elements (frames) simultaneously. Interestingly, as part of their method, Ho et al. [25] introduce "reconstruction guidance" for conditional sampling, which can be seen as a special case of our likelihood approximation (2.15) where the observation y is a subset of x. Lastly, Ho et al. [27] generate low-frame rate, low-resolution videos which are then up-sampled temporally and spatially with a cascade [26] of super-resolution diffusion models. The application of this approach to data assimilation could be worth exploring, although the introduction of arbitrary observation processes seems challenging.

Acknowledgments

François Rozet is a research fellow of the F.R.S.-FNRS (Belgium) and acknowledges its financial support.

References

[1] Gely P. Basharin, Amy N. Langville, and Valeriy A. Naumov. "The life and work of A.A. Markov". In *Linear Algebra and its Applications*. Special Issue on the Conference on the Numerical Solution of Markov Chains 386 (2004).

[2] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[3] A. C. Lorenc. "Analysis methods for numerical weather prediction". In *Quarterly Journal of the Royal Meteorological Society* 112.474 (1986).

[4] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[5] Geir Evensen. "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In *Journal of Geophysical Research: Oceans* 99.C5 (1994).

[6] Thomas M. Hamill. "Ensemble-based atmospheric data assimilation". In *Predictability of Weather and Climate*. 2006.

[7] Yannick Trémolet. "Accounting for an imperfect model in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).

[8] Yannick Trémolet. "Model-error estimation in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 133.626 (2007).

[9] Mike Fisher et al. "Weak-constraint and long window 4DVAR". Tech. rep. ECMWF, 2011.

[10] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[11] ECMWF. "IFS documentation CY47R1 - part II: Data assimilation". In *IFS Documentation CY47R1*. ECMWF, 2020.

[12] Peter Bauer, Alan Thorpe, and Gilbert Brunet. "The quiet revolution of numerical weather prediction". In *Nature* 525.7567 (2015).

[13] Nils Gustafsson et al. "Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres". In *Quarterly Journal of the Royal Meteorological Society* 144.713 (2018).

[14] Julian Mack et al. "Attention-based Convolutional Autoencoders for 3D-Variational Data Assimilation". In *Computer Methods in Applied Mechanics and Engineering* 372 (2020).

[15] Rossella Arcucci et al. "Deep Data Assimilation: Integrating Deep Learning with Data Assimilation". In *Applied Sciences* 11.3 (2021).

[16] Thomas Frerix et al. "Variational Data Assimilation with a Learned Inverse Observation Operator". In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[17] R. Fablet et al. "Learning Variational Data Assimilation Models and Solvers". In *Journal of Advances in Modeling Earth Systems* 13.10 (2021).

[18] Yu-Hong Yeung, David A. Barajas-Solano, and Alexandre M. Tartakovsky. "Physics-Informed Machine Learning Method for Large-Scale Data Assimilation Problems". In *Water Resources Research* 58.5 (2022).

[19] Julien Brajard et al. “Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model”. In *Journal of Computational Science* 44 (2020).

[20] Julien Brajard et al. “Combining data assimilation and machine learning to infer unresolved scale parametrization”. In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (2021).

[21] Kai Zhang et al. “Multi-source information fused generative adversarial network model and data assimilation based history matching for reservoir with complex geologies”. In *Petroleum Science* 19.2 (2022).

[22] Robin Rombach et al. “High-Resolution Image Synthesis With Latent Diffusion Models”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[23] Aditya Ramesh et al. “Hierarchical Text-Conditional Image Generation with CLIP Latents”. 2022.

[24] Chitwan Saharia et al. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”. In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[25] Jonathan Ho et al. “Video Diffusion Models”. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*. 2022.

[26] Jonathan Ho et al. “Cascaded Diffusion Models for High Fidelity Image Generation”. In *Journal of Machine Learning Research* (2022).

[27] Jonathan Ho et al. “Imagen Video: High Definition Video Generation with Diffusion Models”. 2022.

[28] Zhifeng Kong et al. “DiffWave: A Versatile Diffusion Model for Audio Synthesis”. In *International Conference on Learning Representations*. 2021.

[29] Karan Goel et al. “It’s Raw! Audio Generation with State-Space Models”. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[30] Yang Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In *International Conference on Learning Representations*. 2021.

[31] Simo Särkkä and Arno Solin. “Applied Stochastic Differential Equations”. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[32] Qinsheng Zhang and Yongxin Chen. “Fast Sampling of Diffusion Models with Exponential Integrator”. In *International Conference on Learning Representations*. 2023.

[33] Yang Song and Stefano Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution”. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[34] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[35] Brian D. O. Anderson. “Reverse-time diffusion equation models”. In *Stochastic Processes and their Applications* 12.3 (1982).

[36] Aapo Hyvärinen. “Estimation of Non-Normalized Statistical Models by Score Matching”. In *Journal of Machine Learning Research* (2005).

[37] Pascal Vincent. “A Connection Between Score Matching and Denoising Autoencoders”. In *Neural Computation* (2011).

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In *International Conference on Learning Representations*. 2021.

[39] Luping Liu et al. "Pseudo Numerical Methods for Diffusion Models on Manifolds". In *International Conference on Learning Representations*. 2022.

[40] Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[41] Xingchao Liu, Chengyue Gong, and Qiang Liu. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In *International Conference on Learning Representations*. 2023.

[42] Bahjat Kawar, Gregory Vaksman, and Michael Elad. "SNIPS: Solving Noisy Inverse Problems Stochastically". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[43] Yang Song et al. "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models". In *International Conference on Learning Representations*. 2022.

[44] Alexandre Adam et al. "Posterior samples of source galaxies in strong gravitational lenses with score-based priors". 2022.

[45] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In *International Conference on Learning Representations*. 2023.

[46] Bradley Efron. "Tweedie's Formula and Selection Bias". In *Journal of the American Statistical Association* (2011).

[47] Kwanyoung Kim and Jong Chul Ye. "Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[48] François Rozet et al. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[49] Xiangming Meng and Yoshiyuki Kabashima. "Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems". 2022.

[50] G. Parisi. "Correlation functions and computer simulations". In *Nuclear Physics B* 180.3 (1981).

[51] Ulf Grenander and Michael I. Miller. "Representations of Knowledge in Complex Systems". In *Journal of the Royal Statistical Society. Series B (Methodological)* 56.4 (1994).

[52] Edward N. Lorenz. "Deterministic Nonperiodic Flow". In *Journal of the Atmospheric Sciences* 20.2 (1963).

[53] Gary J. Chandler and Rich R. Kerswell. "Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow". In *Journal of Fluid Mechanics* 722 (2013).

[54] Jun S. Liu and Rong Chen. "Sequential Monte Carlo Methods for Dynamic Systems". In *Journal of the American Statistical Association* 93.443 (1998).

[55] Arnaud Doucet and Adam M. Johansen. "A tutorial on particle filtering and smoothing : fiteen years later". In Oxford University Press, 2011.

[56] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In *IEE Proceedings F (Radar and Signal Processing)* 140.2 (1993).

[57] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved Denoising Diffusion Probabilistic Models". In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[58] Dmitrii Kochkov et al. "Machine learning–accelerated computational fluid dynamics". In *Proceedings of the National Academy of Sciences* 118.21 (2021).

[59] Guido Boffetta and Robert E. Ecke. "Two-Dimensional Turbulence". In *Annual Review of Fluid Mechanics* 44.Volume 44, 2012 (2012).

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In *Medical Image Computing and Computer-Assisted Intervention*. 2015.

[61] Yang Song et al. "Consistency Models". In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[62] Evan Archer et al. "Black box variational inference for state space models". 2015.

[63] Chris J Maddison et al. "Filtering Variational Objectives". In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[64] Christian Naesseth et al. "Variational Sequential Monte Carlo". In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. PMLR, 2018.

[65] Dieterich Lawson et al. "SIXO: Smoothing Inference with Twisted Objectives". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[66] Rudolf E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In *Journal of Basic Engineering* 82.1 (1960).

[67] E.A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*. 2000.

[68] Ienkaran Arasaratnam and Simon Haykin. "Cubature Kalman Filters". In *IEEE Transactions on Automatic Control* 54.6 (2009).

[69] Rahul G. Krishnan, Uri Shalit, and David Sontag. "Deep Kalman Filters". 2015.

[70] Laurent Girin et al. "Dynamical Variational Autoencoders: A Comprehensive Review". In *Foundations and Trends in Machine Learning* 15.1-2 (2021).

[71] Suman Ravuri et al. "Skilful precipitation nowcasting using deep generative models of radar". In *Nature* 597.7878 (2021).

[72] Remi Lam et al. "Learning skillful medium-range global weather forecasting". In *Science* 382.6677 (2023).

[73] Kaiming He et al. "Deep Residual Learning for Image Recognition". In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[74] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In *Neural Networks*. Special issue on deep reinforcement learning 107 (2018).

[75] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". 2016.

[76] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In *International Conference on Learning Representations*. 2019.

2.A Unbiased pseudo-blanket approximation

For any continuous random variables a, b and c,

$$\begin{aligned}\nabla_a \log p(a,b) &= \frac{1}{p(a,b)} \nabla_a\, p(a,b) \\ &= \frac{1}{p(a,b)} \nabla_a \int p(a,b,c)\,\mathrm{d}c \\ &= \int \frac{p(c \mid a,b)}{p(a,b,c)} \nabla_a\, p(a,b,c)\,\mathrm{d}c \\ &= \mathbb{E}_{p(c|a,b)}\left[\nabla_a \log p(a,b,c)\right].\end{aligned}$$

Replacing a, b and c by x_t^i, $x_t^{\bar{b}_i}$ and $x_t^{\notin} = \{x_t^j \,:\, j \neq i \wedge j \notin \bar{b}_i\}$, respectively, we obtain

$$\nabla_{x_t^i} \log p(x_t^i, x_t^{\bar{b}_i}) = \mathbb{E}_{p(x_t^{\notin} | x_t^i, x_t^{\bar{b}_i})}\left[\nabla_{x_t^i} \log p(x_t^{1:L})\right],$$

meaning that $\nabla_{x_t^i} \log p(x_t^i, x_t^{\bar{b}_i})$ is the expected value of $\nabla_{x_t^i} \log p(x_t^{1:L})$ over $x_t^{\notin}$. In other words, regardless of the elements or the size of $\bar{b}_i$, $\nabla_{x_t^i} \log p(x_t^i, x_t^{\bar{b}_i})$ is an unbiased (exact in expectation) estimate of $\nabla_{x_t^i} \log p(x_t^{1:L})$.

2.B On the covariance of $p(x \mid x_t)$

Assuming a Gaussian prior $p(x)$ with covariance Σ_x, the covariance $\hat{\Sigma}$ of $p(x \mid x_t)$ takes the form

$$\begin{aligned}\hat{\Sigma} &= \Sigma_x - \Sigma_x \left(\Sigma_x + \frac{\sigma_t^2}{\alpha_t^2} I\right)^{-1} \Sigma_x \\ &= \frac{\sigma_t^2}{\alpha_t^2} Q\Lambda \left(\Lambda + \frac{\sigma_t^2}{\alpha_t^2} I\right)^{-1} Q^{-1}\end{aligned}$$

where $Q\Lambda Q^{-1}$ is the eigendecomposition of Σ_x. We observe that for most of the perturbation time t, the central diagonal term is close to $\Lambda(\Lambda + I)^{-1}$. We therefore propose the covariance approximation

$$\hat{\Sigma} = \frac{\sigma_t^2}{\alpha_t^2} \Gamma$$

where $\Gamma = Q\Lambda\,(\Lambda + I)^{-1}\, Q^{-1}$ is a positive semi-definite matrix.

2.C Algorithms

Algorithm 3 Estimating the posterior score $\nabla_{x_t} \log p(x_t \mid y)$

function $s_\phi(x_t, t \mid y)$
  $s_x \leftarrow s_\phi(x_t, t)$
  $\hat{x} \leftarrow \frac{x_t + \sigma_t^2 \, s_x}{\alpha_t}$
  $s_y \leftarrow \nabla_{x_t} \log \mathcal{N}\left(y \mid \mathcal{A}(\hat{x}), \Sigma_y + \frac{\sigma_t^2}{\alpha_t^2}\Gamma\right)$
  return $s_x + s_y$

Algorithm 4 Predictor-Corrector sampling from $s_\phi(x_t, t \mid y)$

function Sample($\{t_i\}_{i=0}^N, C, \tau$)
  $x_1 \sim \mathcal{N}(0, \Sigma_1)$
  for $i = N$ to 1 **do**
    $x_{t_{i-1}} \leftarrow \frac{\alpha_{t_{i-1}}}{\alpha_{t_i}} x_{t_i} + \left(\frac{\alpha_{t_{i-1}}}{\alpha_{t_i}} - \frac{\sigma_{t_{i-1}}}{\sigma_{t_i}}\right) \sigma_{t_i}^2 \, s_\phi(x_{t_i}, t_i \mid y)$
    for $j = 1$ to C **do**
      $\epsilon \sim \mathcal{N}(0, I)$
      $s \leftarrow s_\phi(x_{t_{i-1}}, t_{i-1} \mid y)$
      $\delta \leftarrow \tau \frac{\dim(s)}{\|s\|_2^2}$
      $x_{t_{i-1}} \leftarrow x_{t_{i-1}} + \delta \, s + \sqrt{2\delta} \, \epsilon$
  return x_0

2.D Experiment details

Resources Experiments were conducted with the help of a cluster of GPUs. In particular, score networks were trained and evaluated concurrently, each on a single GPU with at least 11 GB of memory.

Lorenz 1963 We consider two score network architectures: fully-connected local score networks for small k ($k \leq 4$) and fully-convolutional score networks for large k ($k > 4$). Residual blocks [73], SiLU [74] activation functions and layer normalization [75] are used for both architecture. The number of blocks in the fully-convolutional architecture controls the value of k. We train all score networks for 1024 epochs with the AdamW [76] optimizer and a linearly decreasing learning rate. Other hyperparameters are provided in Table 2.1.

Table 2.1. Score network hyperparameters for the Lorenz experiment.

Hyperparameter	$k \leq 4$	$k > 4$
Architecture	fully-connected	fully-convolutional
Residual blocks	5	$k - 2$
Features/channels	256	64
Kernel size	–	3
Activation	SiLU	SiLU
Normalization	LayerNorm	LayerNorm
Optimizer	AdamW	AdamW
Weight decay	10^{-3}	10^{-3}
Learning rate	10^{-3}	10^{-3}
Scheduler	linear	linear
Epochs	1024	1024
Batches per epoch	256	256
Batch size	256	64

Kolmogorov flow The local score network is a U-Net [60] with residual blocks [73], SiLU [74] activation functions and layer normalization [75]. This architecture is motivated by the locality of the Navier-Stokes equations, which impose that the evolution of a drop of fluid is determined by its immediate environment. This can be seen as an application of the pseudo-blanket approximation (2.13) to a grid-structured set of random variables. We train the score network for 1024 epochs with the AdamW [76] optimizer and a linearly decreasing learning rate. Other hyperparameters are provided in Table 2.2.

Table 2.2. Score network hyperparameters for the Kolmogorov experiment.

Architecture	U-Net
Residual blocks per level	$(3, 3, 3)$
Channels per level	$(96, 192, 384)$
Kernel size	3
Padding	circular
Activation	SiLU
Normalization	LayerNorm
Optimizer	AdamW
Weight decay	10^{-3}
Learning rate	2×10^{-4}
Scheduler	linear
Epochs	1024
Batches per epoch	128
Batch size	32

2.E Assimilation examples

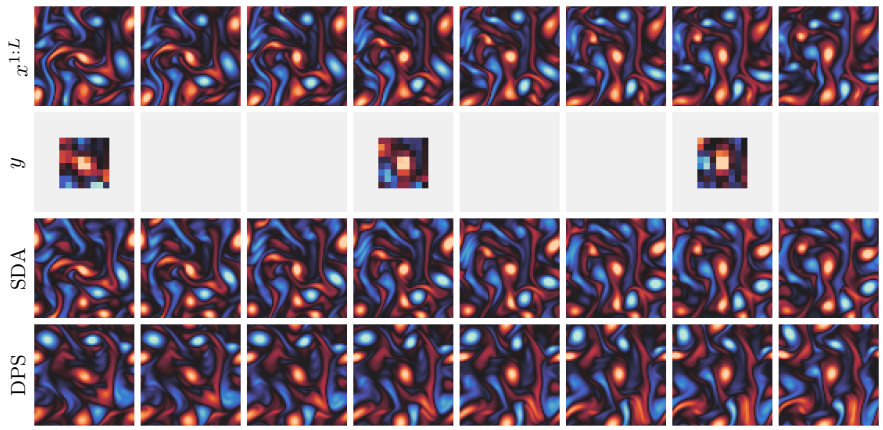


Figure 2.6. Example of sampled trajectory when the observation is insufficient to identify the original trajectory. The observation is the center of the velocity field every three steps, coarsened to a resolution 8×8 and perturbed by a small Gaussian noise ($\Sigma_y = 0.01^2 I$). SDA ($C = 1$, $\tau = 0.5$) identifies the original state where data is sufficient, while generating physically plausible states elsewhere. Replacing SDA's likelihood score approximation with the one of DPS [45] yields a trajectory less consistent with the observation.

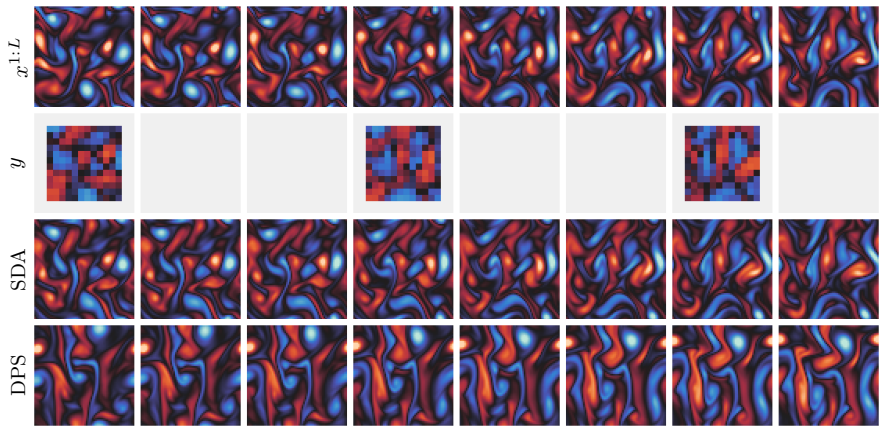


Figure 2.7. Example of sampled trajectory when the observation process is non-linear. The observation corresponds to a saturating transformation $x \mapsto \frac{x}{1+|x|}$ of the vorticity field ω. SDA (512 discretization steps, $C = 1$, $\tau = 0.5$) identifies the original state where data is sufficient, while generating physically plausible states elsewhere. Replacing SDA's likelihood score approximation with the one of DPS [45] yields a trajectory inconsistent with the observation.

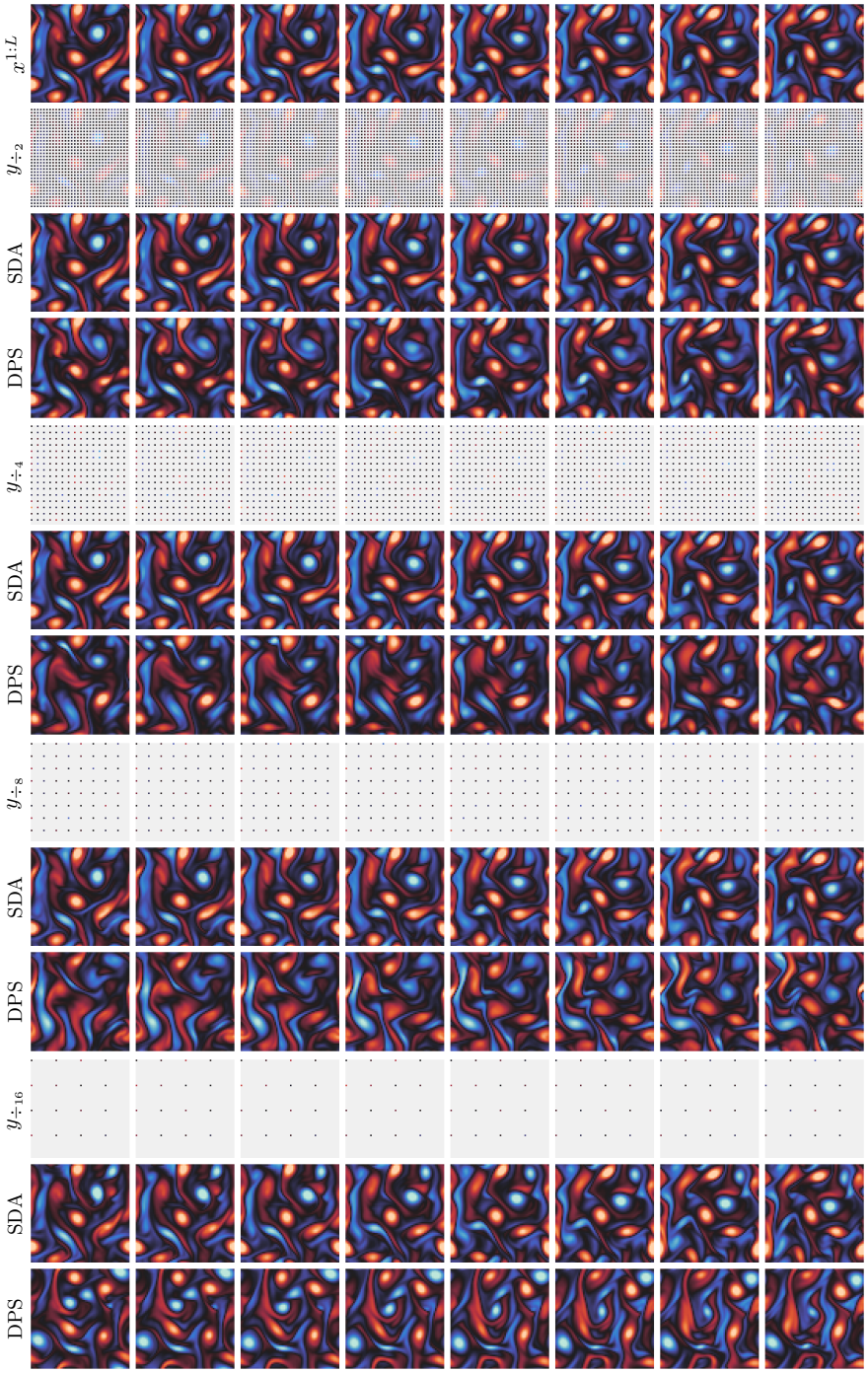


Figure 2.8. Example of sampled trajectories when the observation is spatially sparse. The observation $y_{\div n}$ corresponds to a spatial subsampling of factor n of the velocity field. SDA ($C = 1$, $\tau = 0.5$) closely recovers the original trajectory for all factors n, despite the limited amount of available data. Replacing SDA's likelihood score approximation with the one of DPS [45] leads to trajectories that are progressively less consistent with the observation as n increases.

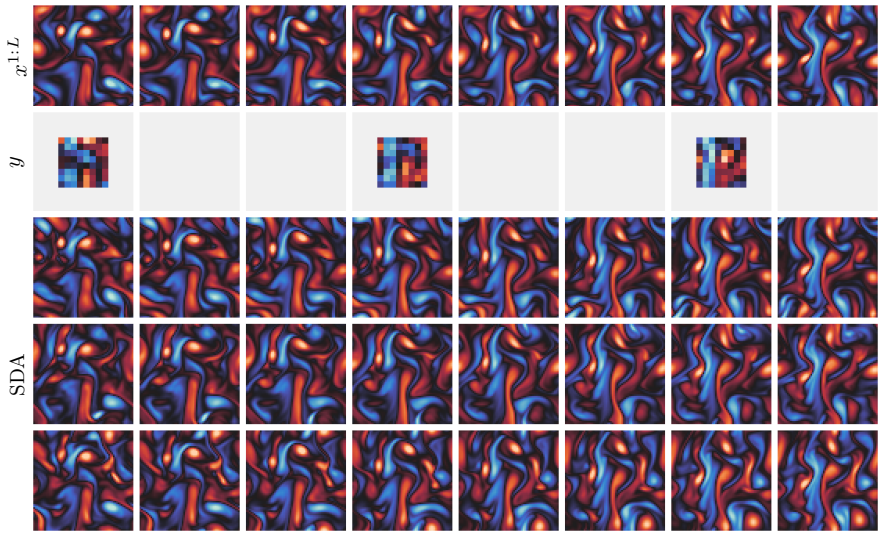


Figure 2.9. Example of sampled trajectories when the observation is insufficient to identify the original trajectory. The observation is the center of the velocity field every three steps, coarsened to a resolution 8×8 and perturbed by a small Gaussian noise ($\Sigma_y = 0.01^2 I$). SDA ($C = 1$, $\tau = 0.5$) identifies the original state where data is sufficient, while generating physically plausible states elsewhere.

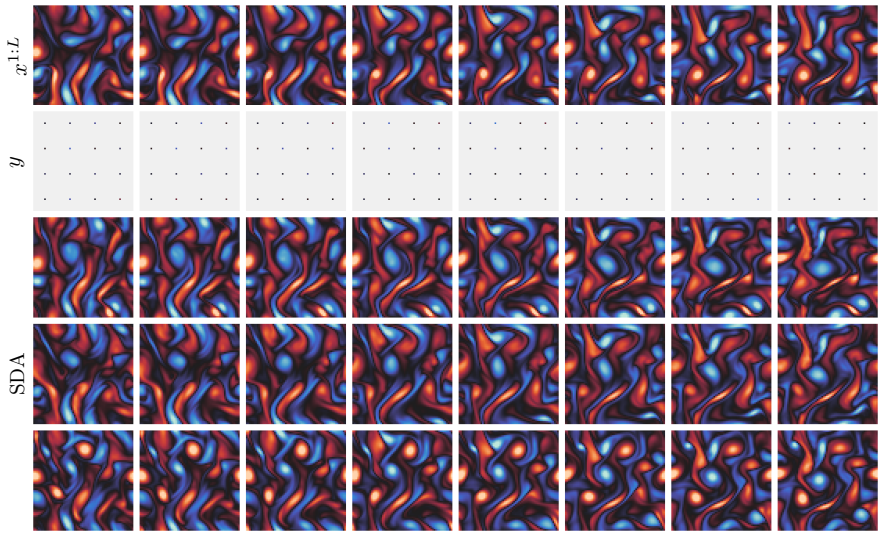


Figure 2.10. Example of sampled trajectories when the observation is spatially sparse. The observation y corresponds to a spatial subsampling of factor 16 of the velocity field with medium Gaussian noise ($\Sigma_y = 0.1^2 I$). SDA ($C = 1$, $\tau = 0.5$) generates trajectories similar to the original, with physically plausible variations.

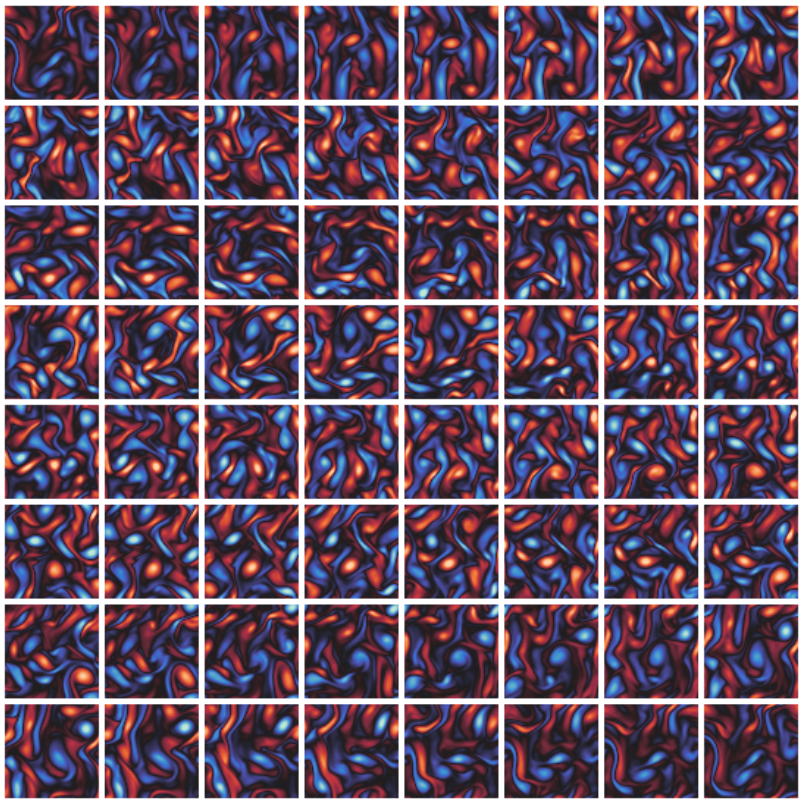


Figure 2.11. Example of long ($L = 127$) sampled trajectory. Odd states are displayed from left to right and from top to bottom. The observation process probes the difference between the initial and final states and the observation is set to zero, which enforces a looping trajectory ($x^1 \approx x^L$). Note that this scenario is not realistic and will therefore lead to physically implausible trajectories.

3 Scaling Score-based Data Assimilation

Turbulence is the most important unsolved problem of classical physics.

— Richard Feynman (1964)

Addendum

This chapter appeared previously as

> François Rozet and Gilles Louppe. "Score-based Data Assimilation for a Two-Layer Quasi-Geostrophic Model". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2023.

As the leading author, François came up with the method, conducted the experiments, interpreted the results, and wrote the manuscript. Gilles supervised the project and suggested experiments.

For the version presented in this chapter, we have slightly edited the mathematical notations from the original publication. Notably, the diffusion time notation $x(t)$ has been replaced by the leaner x_t, which is more common in nowadays literature and more consistent with the rest of this document. The text itself remains unchanged.

Abstract

Data assimilation addresses the problem of identifying plausible state trajectories of dynamical systems given noisy or incomplete observations. In geosciences, it presents challenges due to the high-dimensionality of geophysical dynamical systems, often exceeding millions of dimensions. This work assesses the scalability of score-based data assimilation (SDA), a novel data assimilation method, in the context of such systems. We propose modifications to the score network architecture aimed at significantly reducing memory consumption and execution time. We demonstrate promising results for a two-layer quasi-geostrophic model. The code for all experiments is made available at https://github.com/francois-rozet/sda.

3.1 Introduction

N-layer quasi-geostrophic (NLQG) models [2] are special cases of the Navier-Stokes equations that have been extensively used to describe the dynamics of oceans and atmospheres. NLQG models consider a stratified fluid of N superimposed layers of constant uniform densities; the layers being stacked according to increasing density. The

state of the fluid is fully described by the potential vorticity field q and the velocity fields (u, v) of each layer. In practice, it is necessary to discretize temporally and spatially to solve/integrate the quasi-geostrophic equations, which leads to physical phenomena occurring at a smaller scale than the numerical resolution to be missed [2–5]. In the long run, neglecting such phenomena can lead to poor simulations. It is therefore common to simulate such models at high temporal and spatial resolutions to ensure the quality of the simulations, but this comes at a significant computational cost.

Although they make (shallow water) assumptions to neglect some terms of the Navier-Stokes equations, NLGQ models effectively capture the characteristic features of turbulent ocean systems, such as jet streams, mesoscale eddies, and ocean currents. As such, they are good candidates for designing and benchmarking data assimilation (DA) [6–12] algorithms. Formally, DA targets the problem of inferring the posterior distribution

$$p(x^{1:L} \mid y) = \frac{p(y \mid x^{1:L})}{p(y)} p(x^1) \prod_{i=1}^{L-1} p(x^{i+1} \mid x^i) \tag{3.1}$$

of discrete-time state trajectories $x^{1:L} = (x^1, x^2, \dots, x^L)$ given an observation y resulting from an observation process $p(y \mid x^{1:L})$. In geosciences, the physical model underlying transition dynamics $p(x^{i+1} \mid x^i)$ is typically well known and the observation process is generally formulated as $y = \mathcal{A}(x^{1:L}) + \eta$, where $\mathcal{A}$ is the measurement function and η is a stochastic additive term that accounts for instrumental noise and systematic uncertainties.

In this setting, the simulation quality is very important as it strongly impacts the relevance of inference results. Unfortunately, high resolutions rapidly become a computational burden for classical DA algorithms, such as variational methods [6–10] that require to repeatedly differentiate through the physical model, and Monte Carlo methods [13–16] that conduct large numbers of simulations.

Conversely, score-based data assimilation (SDA) [17], a novel DA method drawing its roots from score-based generative modeling, only relies on the physical model to generate training data. After training, inference can be carried out without relying on the model and at lower temporal and spatial resolutions. However, applying SDA to (very) high-dimensional systems remains an engineering challenge, which we attempt to address in this work.

3.2 Background

Score-based generative modeling has shown remarkable capabilities, powering many of the latest advances in image, video and audio generation [18–22]. In this section, we shortly review score-based generative models and outline how they can be used for solving data assimilation problems with SDA [17].

Continuous-time score-based generative models Adopting the notations of Rozet et al. [17], samples x drawn from a distribution $p(x)$ are progressively perturbed through a continuous-time stochastic diffusion process. This process determines a series of marginal distributions $p(x_t)$ from $t = 0$ to $t = 1$ for which $p(x_0) \approx p(x)$ (clean data) and $p(x_1) \approx \mathcal{N}(0, I)$ (pure noise). By design, the diffusion process can be "reversed" in order to transform noise $x_1 \sim \mathcal{N}(0, I)$ into data samples $x_0 \sim p(x_0)$. However, this reverse process requires access to the quantity $\nabla_{x_t} \log p(x_t)$ known as the score of $p(x_t)$.

A score network is a neural network $s_\phi(x_t, t)$ trained – usually by denoising score matching [23, 24] – to approximate the score $\nabla_{x_t} \log p(x_t)$. After training, a score network can be plugged into the reverse process as a substitute for the true score in order to

generate data samples from $p(x_0)$. A handy property of score-based models is that, under some assumptions on the likelihood $p(y \mid x)$, it is possible to approximate the posterior score

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y \mid x_t) \tag{3.2}$$

without retraining the score network $s_\phi(x_t, t)$, and hence to generate data samples from $p(x_0 \mid y)$.

Score-based data assimilation In score-based data assimilation, the data we try to model are trajectories $x^{1:L}$ and we therefore approximate the trajectory score $\nabla_{x_t^{1:L}} \log p(x_t^{1:L})$. Rozet et al. [17] show that an element $\nabla_{x_t^i} \log p(x_t^{1:L})$ of the trajectory score can be approximated locally by $\nabla_{x_t^i} \log p(x_t^{i-k:i+k})$ for some $k \geq 1$. They therefore propose to train a local score network $s_\phi(x_t^{i-k:i+k}, t)$ to approximate the score over short segments, and compose this local score at inference to generate arbitrarily long trajectories. In addition, the authors introduce a novel approximation for the likelihood score $\nabla_{x_t^{1:L}} \log p(y \mid x_t^{1:L})$ in the assumption of a Gaussian observation process $p(y \mid x^{1:L}) = \mathcal{N}(y \mid \mathcal{A}(x^{1:L}), \Sigma_y)$, which covers many observation scenarios. Finally, to prevent approximation errors from accumulating along the simulation of the reverse process, Rozet et al. [17] perform a few Langevin Monte Carlo corrections between each step of the discretized reverse process.

3.3 Task

Although the Kolmogorov system presented in the experiments conducted by Rozet et al. [17] is high-dimensional (tens of thousands of dimensions) compared to what is approachable with classical posterior inference methods, it remains small in comparison to the millions of dimensions of some operational DA systems.

In this study, we attempt to perform data assimilation for a two-layer quasi-geostrophic (2LQG) model. We choose a torus-like 1000 km periodic domain, discretized into 256×256 spatial bins. The state is described by the potential vorticity field q^l and the zonal and meridional velocity fields u^l and v^l for each layer $l \in \{1, 2\}$. A state snapshot $x = (q_1, u_1, v_1, q_2, u_2, v_2)$ is therefore a 256×256 grid with 6 channels, which exceeds the dimensionality of Rozet et al. [17]'s Kolmogorov system by a factor 48.

We use the `pyqg` [25] Python library to solve the quasi-geostrophic equations using pseudo-spectral methods. To ensure the quality of the simulation, the domain is discretized into 512×512 spatial bins and the integration timestep is set to 15 min. We coarsen the simulation to the target resolution (256×256) afterwards and separate two snapshots x^i and x^{i+1} by a day (24 h). To ensure that the system has reached statistical stationarity, that is $p(x^i) = p(x^{i+1})$, it is simulated for 5 years starting from random initial conditions before saving the first state x^1 [3–5].

3.4 Architecture

The high-dimensionality of the task at hand (and operational DA systems) introduces many engineering challenges. Notably, naively scaling the U-Net [26] inspired score network architecture proposed by Rozet et al. [17] to our task results in memory consumption exceeding the total memory of a A5000 GPU (24 GB) when training on segments, which gets worse at inference on full trajectories. We propose several modifications to the architecture to address these problems, avoiding low-level tricks such as custom CUDA kernels.

First, at inference, except for the extremities of the trajectory, only the $k+1$-th element of the local score network's output is actually used. The other elements are therefore wasting both memory and compute resources. Instead, we make the score network fully-convolutional in the time axis, such that it can be applied to trajectories of any length. However, we limit its temporal receptive field such that it can still be trained on short segments. To do so, we make all layers 2-D spatial convolutions (no mixing along the time axis) and strategically add a few ($2k$) 1-D temporal convolutions. Assuming the same number of channels per state, this modification roughly reduces the amount of memory consumed and computation performed at inference by a factor $2k+1$.

Still, training on segments of 256×256 grid states consumes a lot of memory. To solve this, we combine two ideas: activation checkpointing and inverted bottleneck blocks. Activation checkpointing [27, 28] is an automatic differentiation trick that consists in only keeping in memory some intermediate values of the computation graph, instead of all. The missing values are recomputed during the backward pass, effectively trading memory consumption for execution time. Inverted bottleneck blocks [29, 30] consist in limiting the number of channels of a convolutional network except within residual blocks where the number of channels is expanded to perform non-linear computations and contracted back afterwards. By checkpointing only the intermediate values with a limited number of channels, we are able to reduce memory consumption by 72 %, while increasing execution time by only 27 %. A diagram of the architecture incorporating the proposed modifications is provided in Figure 3.2.

3.5 Results

Following Rozet et al. [17], we choose a variance preserving diffusion process for our experiments and simulate the reverse process in 256 evenly spaced discretization steps and one Langevin Monte Carlo correction per step. The score networks are trained once and then evaluated under various observation scenarios.

As described in Section 3.4, the local score network presents a U-Net [26] inspired architecture with efficiency-oriented modifications. The temporal receptive field of the score network is such that $k = 4$ and, therefore, it can be trained on segments of length $2k + 1 = 9$ instead of entire trajectories. Using `pyqg` [25], we generate 1024 trajectories of 32 state snapshots, which are split into training (80 %), validation (10 %) and evaluation (10 %) sets. For practical reasons, the vorticity and velocity fields (channels) are standardized to have unit variance. We train the score network for 1024 epochs (48 h on a single A5000 GPU) with the AdamW [31] optimizer and a linearly decreasing learning rate. Further architecture and training details are provided in Appendix 3.A.

We consider a scenario where weather stations placed regularly over the domain measure the local velocities (u_1, v_1, u_2, v_2) every day for a month ($L = 32$). We generate an observation y for a trajectory of the evaluation set. Given the observation, we sample three trajectories with SDA and find that they closely resembles the original trajectory, as illustrated in Figure 3.1. A similar experiment where weather stations are placed randomly upon the domain and with more measurement noise is presented in Figure 3.3. In both cases, however, we observe that the generated trajectories exhibit less small-scale details than a simulation, which indicates that the score network does not model the data perfectly. There are several avenues to explore in order to address this issue, such as training for longer with more data or increasing the capacity (more channels or layers) of the score network. Fourier neural operators (FNOs) could be another way to make the network more expressive and have proved to be effective for emulating partial differential equations [5, 32, 33].

Importantly, the three trajectories in Figure 3.1 were generated concurrently on a single

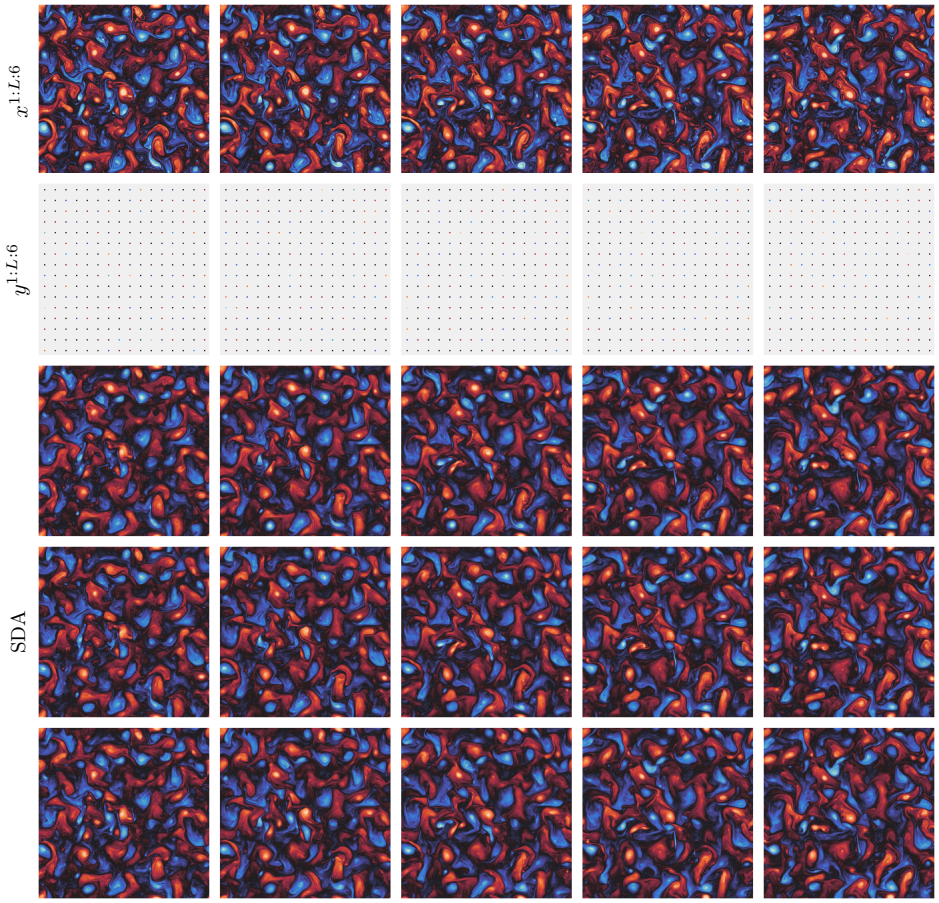


Figure 3.1. Example of sampled trajectories for a spatially sparse observation. States are visualized by their potential vorticity field q. Positive values (red) indicate clockwise rotation and negative values (blue) indicate counter-clockwise rotation. The observation y corresponds to a spatial subsampling of factor 16 of the velocity fields (u_1, v_1, u_2, v_2) with small Gaussian noise ($\Sigma_y = 0.01^2 I$). SDA generates trajectories similar to the original one, despite the limited amount of information available in the observation. The three trajectories present slight physically plausible variations, as expected from sampling from a narrow posterior. We observe that the trajectories exhibit less small-scale details than the original one.

A5000 GPU in 18 minutes and using 14 GB of memory. In comparison, naively scaling the original architecture of Rozet et al. [17] would have required roughly 10 times more compute and 20 times more memory. With classical data assimilation algorithms, such as variational [6–10] or Monte Carlo [13–16] methods, we estimate that inference would have required the compute equivalent of thousands of simulations, each taking roughly 3 minutes to complete using `pyqg` [25] on 4 CPU cores. In conclusion, we believe that SDA is ready to be tested in full operational conditions.

Acknowledgments

François Rozet is a research fellow of the F.R.S.-FNRS (Belgium) and acknowledges its financial support.

References

[1] François Rozet and Gilles Louppe. "Score-based Data Assimilation for a Two-Layer Quasi-Geostrophic Model". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2023.

[2] Colin Cotter et al. "Data Assimilation for a Quasi-Geostrophic Model with Circulation-Preserving Stochastic Transport Noise". In *Journal of Statistical Physics* 179.5 (2020).

[3] Thomas Bolton and Laure Zanna. "Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization". In *Journal of Advances in Modeling Earth Systems* 11.1 (2019).

[4] Andrew Ross et al. "Benchmarking of Machine Learning Ocean Subgrid Parameterizations in an Idealized Model". In *Journal of Advances in Modeling Earth Systems* 15.1 (2023).

[5] Victor Mangeleer and Gilles Louppe. "Ocean parameterizations in an idealized model using machine learning". PhD thesis. Université de Liège, 2023.

[6] A. C. Lorenc. "Analysis methods for numerical weather prediction". In *Quarterly Journal of the Royal Meteorological Society* 112.474 (1986).

[7] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[8] Yannick Trémolet. "Accounting for an imperfect model in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).

[9] Yannick Trémolet. "Model-error estimation in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 133.626 (2007).

[10] Mike Fisher et al. "Weak-constraint and long window 4DVAR". Tech. rep. ECMWF, 2011.

[11] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[12] ECMWF. "IFS documentation CY47R1 - part II: Data assimilation". In *IFS Documentation CY47R1*. ECMWF, 2020.

[13] Geir Evensen. "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In *Journal of Geophysical Research: Oceans* 99.C5 (1994).

[14] Jun S. Liu and Rong Chen. "Sequential Monte Carlo Methods for Dynamic Systems". In *Journal of the American Statistical Association* 93.443 (1998).

[15] Thomas M. Hamill. "Ensemble-based atmospheric data assimilation". In *Predictability of Weather and Climate*. 2006.

[16] Arnaud Doucet and Adam M. Johansen. "A tutorial on particle filtering and smoothing : fiteen years later". In Oxford University Press, 2011.

[17] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[18] Robin Rombach et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[19] Aditya Ramesh et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents". 2022.

[20] Chitwan Saharia et al. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”. In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[21] Jonathan Ho et al. “Video Diffusion Models”. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*. 2022.

[22] Karan Goel et al. “It’s Raw! Audio Generation with State-Space Models”. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[23] Aapo Hyvärinen. “Estimation of Non-Normalized Statistical Models by Score Matching”. In *Journal of Machine Learning Research* (2005).

[24] Pascal Vincent. “A Connection Between Score Matching and Denoising Autoencoders”. In *Neural Computation* (2011).

[25] Ryan Abernathey et al. “pyqg: Python Quasigeostrophic Model”. 2022.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In *Medical Image Computing and Computer-Assisted Intervention*. 2015.

[27] Tianqi Chen et al. “Training Deep Nets with Sublinear Memory Cost”. 2016.

[28] Audrunas Gruslys et al. “Memory-Efficient Backpropagation Through Time”. In *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[29] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

[30] Zhuang Liu et al. “A ConvNet for the 2020s”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[31] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In *International Conference on Learning Representations*. 2019.

[32] Zongyi Li et al. “Fourier Neural Operator for Parametric Partial Differential Equations”. In *International Conference on Learning Representations*. 2021.

[33] Alasdair Tran et al. “Factorized Fourier Neural Operators”. In *International Conference on Learning Representations*. 2023.

[34] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[35] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”. In *Neural Networks*. Special issue on deep reinforcement learning 107 (2018).

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. “Layer Normalization”. 2016.

3.A Experiment details

Resources Experiments were conducted with the help of a cluster of GPUs. In particular, score networks were trained and evaluated concurrently, each on a single GPU with 24 GB of memory.

Score network The local score network is a U-Net [26] with inverted bottleneck residual blocks [29, 30, 34], SiLU [35] activation functions and layer normalization [36]. Residual blocks either operate on the temporal axis or the spatial axes (see Section 3.4). Gradient checkpointing [27, 28] is applied to the residual blocks to reduce memory consumption. Other hyperparameters are provided in Table 3.1 and a schematic representation of the architecture is provided in Figure 3.2.

Table 3.1. Score network hyperparameters for the 2LQG experiment.

Spatial blocks per level	$(3, 3, 3)$
Channels per level	$(16, 32, 64)$
Inverted bottleneck factor	4
Kernel size	5
Padding	circular
Activation	SiLU
Normalization	LayerNorm
Optimizer	AdamW
Weight decay	10^{-3}
Learning rate	2×10^{-4}
Scheduler	linear
Epochs	1024
Batch size	4

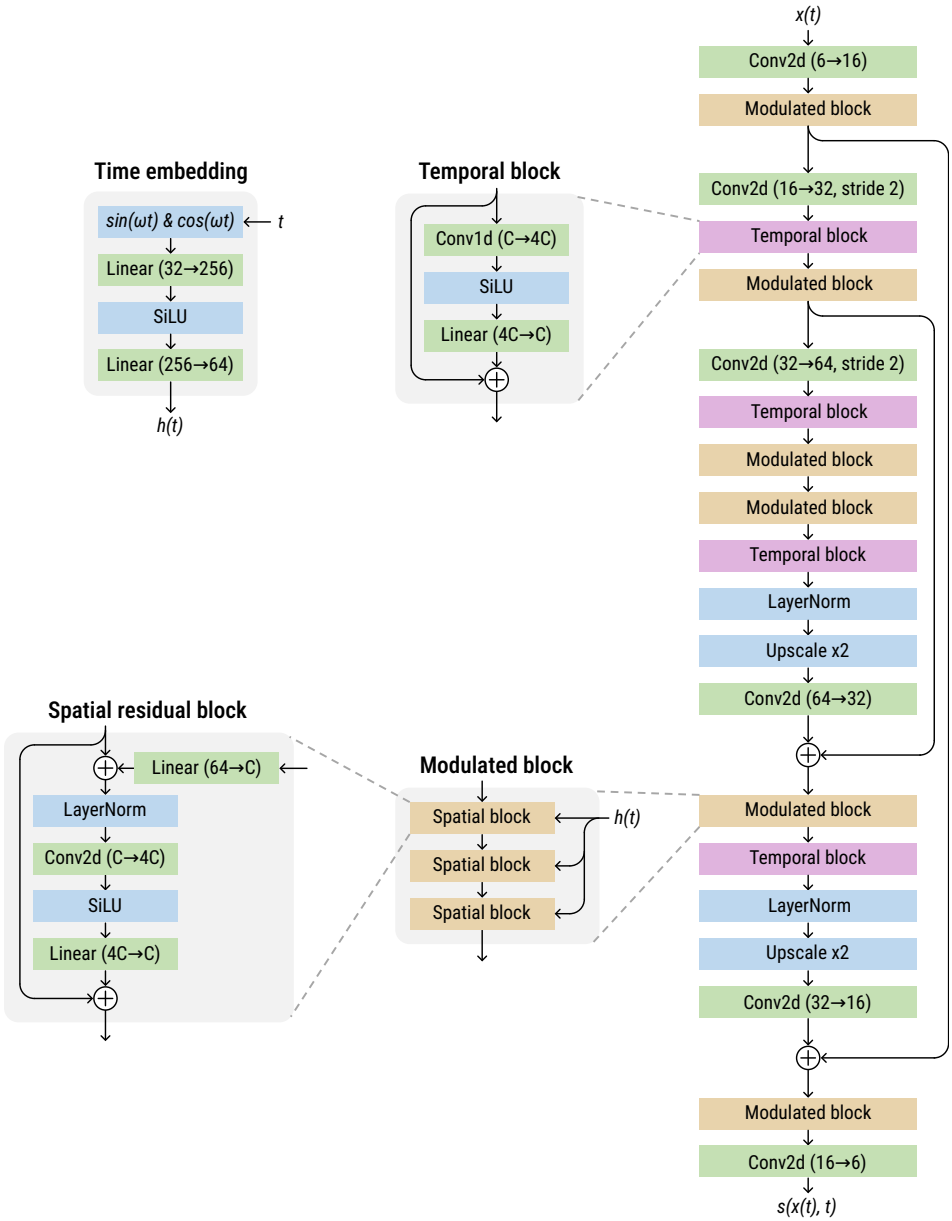


Figure 3.2. Schematic representation of the score network architecture. All spatial and temporal blocks are gradient checkpointed to reduce memory consumption.

3.B Assimilation examples

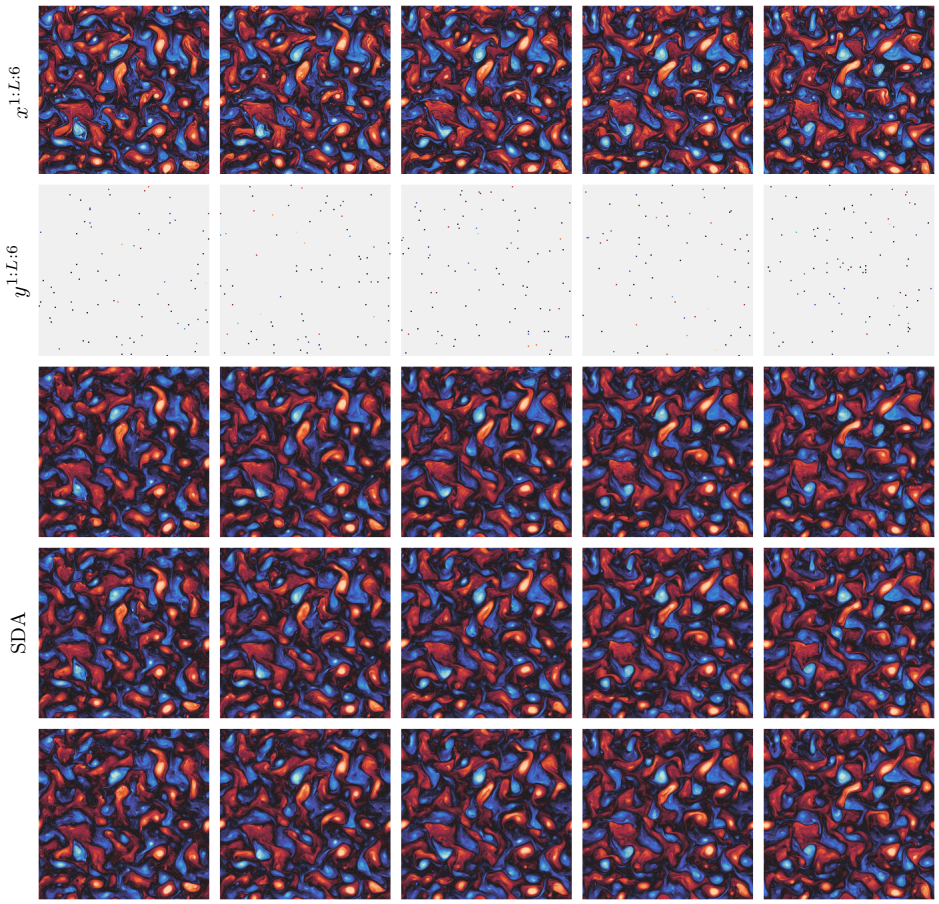


Figure 3.3. Example of sampled trajectories for a spatially sparse observation. The observation y corresponds to a random (uniform) sampling of ±80 bins of the velocity fields (u_1, v_1, u_2, v_2) with medium Gaussian noise ($\Sigma_y = 0.1^2 I$). SDA generates trajectories similar to the original one, despite the limited amount of information available in the observation. The three trajectories present slight physically plausible variations, as expected from sampling from a narrow posterior. We observe that the trajectories exhibit less small-scale details than the original one.

4 Learning Diffusion Priors by Expectation Maximization

What we observe is not nature herself, but nature exposed to our method of questioning.

— Werner Heisenberg (1958)

Addendum

This chapter has previously been published as

> François Rozet, Gérôme Andry, François Lanusse, and Gilles Louppe. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

The goal of this project, training diffusion models from incomplete and/or noisy data, was proposed by François L. and Gilles. As the leading author, François R. came up with the methods, conducted the experiments, interpreted the results, and wrote the manuscript. Gilles supervised the project, established the link with the EM algorithm, suggested experiments, and participated in the writing and literature review. Gérôme reviewed the manuscript and contributed to the methods through our frequent conversations.

For the version presented in this chapter, we have slightly modified the placement of figures with respect to the original publication. We have also given a name to the main method: diffusion-based expectation-maximization or DiEM, for short. The content remains otherwise unchanged.

Abstract

Diffusion models recently proved to be remarkable priors for Bayesian inverse problems. However, training these models typically requires access to large amounts of clean data, which could prove difficult in some settings. In this work, we present DiEM, a novel method based on the expectation-maximization algorithm for training diffusion models from incomplete and noisy observations only. Unlike previous works, DiEM leads to proper diffusion models, which is crucial for downstream tasks. As part of our methods, we propose and motivate an improved posterior sampling scheme for unconditional diffusion models. We present empirical evidence supporting the effectiveness of our approach.

4.1 Introduction

Many scientific applications can be formalized as Bayesian inference in latent variable models, where the target is the posterior distribution $p(x \mid y) \propto p(y \mid x)\, p(x)$ given an observation $y \in \mathbb{R}^M$ resulting from a forward process $p(y \mid x)$ and a prior distribution $p(x)$ over the latent variable $x \in \mathbb{R}^N$. Notable examples include gravitational lensing inversion [3–5], accelerated MRI [6–10], unfolding in particle physics [11, 12], and data assimilation [13–16]. In all of these examples, the observation y alone is either too incomplete or too noisy to recover the latent x. Additional knowledge in the form of an informative prior $p(x)$ is crucial for valuable inference.

Recently, diffusion models [17, 18] proved to be remarkable priors for Bayesian inference, demonstrating both quality and versatility [19–29]. However, to train a diffusion model for the latent x, one would typically need a large number of latent realizations, which by definition are not or rarely accessible. This is notably the case in earth and space sciences where the systems of interest can only be probed superficially.

Empirical Bayes (EB) methods [30–33] offer a solution to the problem of prior specification in latent variable models when only observations y are available. The objective of EB is to find the parameters θ of a prior model $q_\theta(x)$ for which the evidence distribution $q_\theta(y) = \int p(y \mid x)\, q_\theta(x)\, \mathrm{d}x$ is closest to the empirical distribution of observations $p(y)$. Many EB methods have been proposed over the years, but they remain limited to low-dimensional settings [34–39] or simple parametric models [40, 41].

In this work, our goal is to use diffusion models for the prior $q_\theta(x)$, as they are best-in-class for modeling high-dimensional distributions and enable many downstream tasks, including Bayesian inference. This presents challenges for previous empirical Bayes methods which typically rely on models for which the density $q_\theta(x)$ or samples $x \sim q_\theta(x)$ are differentiable with respect to the parameters θ. Instead, we propose DiEM, an adaptation of the expectation-maximization [42–46] algorithm where we alternate between generating samples from the posterior $q_\theta(x \mid y)$ and training the prior $q_\theta(x)$ on these samples. As part of our method, we propose an improved posterior sampling scheme for unconditional diffusion models, which we motivate theoretically and empirically.

4.2 Diffusion Models

The primary purpose of diffusion models (DMs) [17, 18], also known as score-based generative models [47, 48], is to generate plausible data from a distribution $p(x)$ of interest. Formally, adapting the continuous-time formulation of Song et al. [48], samples $x \in \mathbb{R}^N$ from $p(x)$ are progressively perturbed through a diffusion process expressed as a stochastic differential equation (SDE)

$$\mathrm{d}x_t = f_t\, x_t\, \mathrm{d}t + g_t\, \mathrm{d}w_t \tag{4.1}$$

where $f_t \in \mathbb{R}$ is the drift coefficient, $g_t \in \mathbb{R}_+$ is the diffusion coefficient, $w_t \in \mathbb{R}^N$ denotes a standard Wiener process and $x_t \in \mathbb{R}^N$ is the perturbed sample at time $t \in [0, 1]$. Because the SDE is linear with respect to x_t, the perturbation kernel from x to x_t is Gaussian and takes the form

$$p(x_t \mid x) = \mathcal{N}(x_t \mid \alpha_t\, x, \Sigma_t) \tag{4.2}$$

where α_t and $\Sigma_t = \sigma_t^2 I$ are derived from f_t and g_t [48–51]. Crucially, the forward SDE (4.1) has an associated family of reverse SDEs [48–51]

$$\mathrm{d}x_t = \left[f_t\, x_t - \frac{1 + \eta^2}{2} g_t^2\, \nabla_{x_t} \log p(x_t) \right] \mathrm{d}t + \eta\, g_t\, \mathrm{d}w_t \tag{4.3}$$

where $\eta \geq 0$ is a parameter controlling stochasticity. In other words, we can draw noise samples $x_1 \sim p(x_1) \approx \mathcal{N}(0, \Sigma_1)$ and gradually remove the noise therein to obtain data samples $x_0 \sim p(x_0) \approx p(x)$ by simulating Eq. (4.3) from $t = 1$ to 0 using an appropriate discretization scheme [18, 47, 48, 51–54]. In this work, we adopt the variance exploding SDE [47] for which $f_t = 0$ and $\alpha_t = 1$.

In practice, the score function $\nabla_{x_t} \log p(x_t)$ in Eq. (4.3) is unknown, but can be approximated by a neural network trained via denoising score matching [55, 56]. Several equivalent parameterizations and objectives have been proposed for this task [18, 47, 48, 52–54]. In this work, we adopt the denoiser parameterization $d_\theta(x_t, t)$ and its objective [53]

$$\arg\min_\theta \mathbb{E}_{p(x)p(t)p(x_t|x)} \left[\lambda_t \left\| d_\theta(x_t, t) - x \right\|_2^2 \right] , \tag{4.4}$$

for which the optimal denoiser is the mean $\mathbb{E}[x \mid x_t]$ of $p(x \mid x_t)$. Importantly, $\mathbb{E}[x \mid x_t]$ is linked to the score function through Tweedie's formula [57–60]

$$\mathbb{E}[x \mid x_t] = x_t + \Sigma_t \nabla_{x_t} \log p(x_t) , \tag{4.5}$$

which allows to use $s_\theta(x_t) = \Sigma_t^{-1}(d_\theta(x_t, t) - x_t)$ as a score estimate in Eq. (4.3).

4.3 Expectation-Maximization

The objective of the expectation-maximization (EM) algorithm [42–46] is to find the parameters θ of a latent variable model $q_\theta(x, y)$ that maximize the log-evidence $\log q_\theta(y)$ of an observation y. For a distribution of observations $p(y)$, the objective is to maximize the expected log-evidence [45, 46] or, equivalently, to minimize the Kullback-Leibler (KL) divergence between $p(y)$ and $q_\theta(y)$. That is,

$$\theta^* = \arg\max_\theta \mathbb{E}_{p(y)} \big[\log q_\theta(y) \big] \tag{4.6}$$

$$= \arg\min_\theta \mathrm{KL}\big(p(y) \,\|\, q_\theta(y) \big) . \tag{4.7}$$

The key idea behind the EM algorithm is that for any two sets of parameters θ_a and θ_b, we have

$$\log \frac{q_{\theta_a}(y)}{q_{\theta_b}(y)} = \log \frac{q_{\theta_a}(x, y)}{q_{\theta_b}(x, y)} + \log \frac{q_{\theta_b}(x \mid y)}{q_{\theta_a}(x \mid y)} \tag{4.8}$$

$$= \mathbb{E}_{q_{\theta_b}(x|y)} \left[\log \frac{q_{\theta_a}(x, y)}{q_{\theta_b}(x, y)} \right] + \mathrm{KL}\big(q_{\theta_b}(x \mid y) \,\|\, q_{\theta_a}(x \mid y) \big) \tag{4.9}$$

$$\geq \mathbb{E}_{q_{\theta_b}(x|y)} \big[\log q_{\theta_a}(x, y) - \log q_{\theta_b}(x, y) \big] . \tag{4.10}$$

This inequality also holds in expectation over $p(y)$. Therefore, starting from arbitrary parameters θ_0, the EM update

$$\theta_{k+1} = \arg\max_\theta \mathbb{E}_{p(y)} \mathbb{E}_{q_{\theta_k}(x|y)} \big[\log q_\theta(x, y) - \log q_{\theta_k}(x, y) \big] \tag{4.11}$$

$$= \arg\max_\theta \mathbb{E}_{p(y)} \mathbb{E}_{q_{\theta_k}(x|y)} \big[\log q_\theta(x, y) \big] \tag{4.12}$$

leads to a sequence of parameters θ_k for which the expected log-evidence $\mathbb{E}_{p(y)} \big[\log q_{\theta_k}(y) \big]$ is monotonically increasing and converges to a local optimum [44–46].

When the expectation in Eq. (4.12) is intractable, many have proposed to use Monte Carlo approximations instead [61–68]. Previous approaches include Markov chain Monte Carlo (MCMC) sampling, importance sampling, rejection sampling and their variations [65–68]. A perhaps surprising advantage of Monte Carlo EM (MCEM) algorithms is that they may be able to overcome local optimum traps [62, 63]. We refer the reader to Ruth [68] for a recent review of MCEM algorithms.

4.4 Methods

Although rarely mentioned in the literature, the expectation-maximization algorithm is a possible solution to the empirical Bayes problem. Indeed, both have the same objective: minimizing the KL between the empirical distribution of observations $p(y)$ and the evidence $q_\theta(y)$. In the empirical Bayes setting, the forward model $p(y \mid x)$ is known and only the parameters of the prior $q_\theta(x)$ should be optimized. In this case, Eq. (4.12) becomes

$$\theta_{k+1} = \arg\max_\theta \mathbb{E}_{p(y)} \mathbb{E}_{q_{\theta_k}(x|y)} \big[\log q_\theta(x) + \log p(y \mid x) \big] \tag{4.13}$$

$$= \arg\max_\theta \mathbb{E}_{p(y)} \mathbb{E}_{q_{\theta_k}(x|y)} \big[\log q_\theta(x) \big] \tag{4.14}$$

$$= \arg\min_\theta \mathrm{KL}\big(\pi_k(x) \,\|\, q_\theta(x)\big) \tag{4.15}$$

where $\pi_k(x) = \int q_{\theta_k}(x \mid y)\, p(y)\, \mathrm{d}y$. Intuitively, $\pi_k(x)$ and therefore $q_{\theta_{k+1}}(x)$ assign more density to latents x which are consistent with observations $y \sim p(y)$ than $q_{\theta_k}(x)$. In this work, we consider a special case of the empirical Bayes problem where each observation y has an associated measurement matrix A and the forward process takes a linear Gaussian form $p(y \mid x, A) = \mathcal{N}(y \mid Ax, \Sigma_y)$. This formulation allows the forward process to be potentially different for each observation y. For example, if the position or environment of a sensor changes, the measurement matrix A may also change, which leads to an empirical distribution of pairs $(y, A) \sim p(y, A)$. As a result, $\pi_k(x)$ in Eq. (4.15) becomes $\pi_k(x) = \int q_{\theta_k}(x \mid y, A)\, p(y, A)\, \mathrm{d}y$.

4.4.1 Diffusion-based Expectation-Maximization

Now that our goals and assumptions are set, we present our method to learn a diffusion model $q_\theta(x)$ for the latent x from observations y by expectation-maximization. The idea is to decompose Eq. (4.15) into (i) generating a dataset of i.i.d. samples from $\pi_k(x)$ and (ii) training $q_{\theta_{k+1}}(x)$ to reproduce the generated dataset. We summarize the pipeline in Algorithms 5, 6 and 7, provided in Appendix 4.A due to space constraints.

Expectation To draw from $\pi_k(x)$, we first sample a pair $(y, A) \sim p(y, A)$ and then generate $x \sim q_{\theta_k}(x \mid y, A)$ from the posterior. Unlike previous MCEM algorithms that rely on expensive and hard to tune sampling strategies [65–68], the use of a diffusion model enables efficient and embarrassingly parallelizable posterior sampling [23–25]. However, the quality of posterior samples is critical for the EM algorithm to converge properly [65–68] and, in this regard, we find previous posterior sampling methods [23–25, 27, 28] to be unsatisfactory. Therefore, we propose an improved posterior sampling scheme, named moment matching posterior sampling (MMPS), which we present and motivate in Section 4.4.2. We evaluate MMPS independently from the context of learning from observations in Appendix 4.E.

Maximization We parameterize our diffusion model $q_\theta(x)$ by a denoiser network $d_\theta(x_t, t)$ and train it via denoising score matching [55, 56], as presented in Section 4.2. To accelerate the training, we start each iteration with the previous parameters θ_k.

Initialization An important part of our pipeline is the initial prior $q_0(x)$. Any initial prior leads to a local optimum [44–46], but an informed initial prior can reduce the number of iterations until convergence. In our experiments, we take a Gaussian distribution $\mathcal{N}(x \mid \mu_x, \Sigma_x)$ as initial prior and fit its mean and covariance by – you guessed it! – expectation-maximization. Fitting a Gaussian distribution by EM is very fast as the

maximization step can be conducted in closed-form, especially for low-rank covariance approximations [69].

An alternative we do not explore in this work would be to use a pre-trained diffusion model as initial prior. Pre-training can be conducted on data we expect to be similar to the latents, such as computer simulations or even video games. The more similar, the faster the EM algorithm converges. However, if the initial prior $q_0(x)$ does not cover latents that are otherwise plausible under the observations, the following priors $q_{\theta_k}(x)$ may not cover these latents either. A conservative initial prior is therefore preferable for scientific applications.

4.4.2 Moment Matching Posterior Sampling

To sample from the posterior distribution $q_\theta(x \mid y) \propto q_\theta(x)\, p(y \mid x)$ of our diffusion prior $q_\theta(x)$, we have to estimate the posterior score $\nabla_{x_t} \log q_\theta(x_t \mid y)$ and plug it into the reverse SDE (4.3). In this section, we propose and motivate an improved approximation for the posterior score. As this contribution is not bound to the context of EM, we temporarily switch back to the notations of Section 4.2 where our prior is denoted $p(x)$ instead of $q_\theta(x)$.

Diffusion posterior sampling Thanks to Bayes' rule, the posterior score $\nabla_{x_t} \log p(x_t \mid y)$ can be decomposed into two terms [19, 20, 23–27, 48]

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y \mid x_t)\,. \tag{4.16}$$

If there be two subsequent events, the probability of the second M/N and the probability of both together P/N, and it being first discovered that the second event has also happened, the probability I am right is $\frac{P}{M}$.

— Thomas Bayes (1763)

As an estimate of the prior score $\nabla_{x_t} \log p(x_t)$ is already available via the denoiser $d_\theta(x_t, t)$, the remaining task is to estimate the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$. Assuming a differentiable measurement function $\mathcal{A}$ and a Gaussian forward process $p(y \mid x) = \mathcal{N}(y \mid \mathcal{A}(x), \Sigma_y)$, Chung et al. [23] propose the approximation

$$p(y \mid x_t) = \int p(y \mid x)\, p(x \mid x_t)\, \mathrm{d}x \approx \mathcal{N}\left(y \mid \mathcal{A}(\mathbb{E}[x \mid x_t]), \Sigma_y\right) \tag{4.17}$$

which allows to estimate the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ without training any other network than $d_\theta(x_t, t) \approx \mathbb{E}[x \mid x_t]$. The motivation behind Eq. (4.17) is that, when σ_t is small, assuming that $p(x \mid x_t)$ is narrowly concentrated around its mean $\mathbb{E}[x \mid x_t]$ is reasonable. However, this approximation is very poor when σ_t is not negligible. Consequently, DPS [23] is unstable, does not properly cover the support of the posterior $p(x \mid y)$ and often leads to samples x which are inconsistent with the observation y [24–27].

Moment matching To address these flaws, following studies [24–27] take the covariance $\mathbb{V}[x \mid x_t]$ into account when estimating the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$. Specifically, they consider the Gaussian approximation

Good approximations often lead to better ones.

— George Pólya (1977)

$$q(x \mid x_t) = \mathcal{N}\left(x \mid \mathbb{E}[x \mid x_t], \mathbb{V}[x \mid x_t]\right) \tag{4.18}$$

which is closest to $p(x \mid x_t)$ in Kullback-Leibler (KL) divergence [71]. Then, assuming a linear Gaussian forward process $p(y \mid x) = \mathcal{N}(y \mid Ax, \Sigma_y)$, we obtain [71]

$$q(y \mid x_t) = \int p(y \mid x)\, q(x \mid x_t)\, \mathrm{d}x = \mathcal{N}\left(y \mid A\mathbb{E}[x \mid x_t], \Sigma_y + A\mathbb{V}[x \mid x_t]A^\top\right) \tag{4.19}$$

which allows to estimate the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ as

$$\nabla_{x_t} \log q(y \mid x_t) = \nabla_{x_t} \mathbb{E}[x \mid x_t]^\top A^\top \left(\Sigma_y + A\mathbb{V}[x \mid x_t]A^\top\right)^{-1} \left(y - A\mathbb{E}[x \mid x_t]\right) \tag{4.20}$$

under the assumption that the derivative of $\mathbb{V}[x \mid x_t]$ with respect to x_t is negligible [26, 27]. Even with simple heuristics for $\mathbb{V}[x \mid x_t]$, such as Σ_t [22] or $(\Sigma_t^{-1} + \Sigma_x^{-1})^{-1}$ [24, 25], this adaptation leads to significantly more stable sampling and better coverage of the posterior $p(x \mid y)$ than DPS [23]. However, we find that heuristics lead to overly dispersed posteriors $q(x_t \mid y) \propto p(x_t)\, q(y \mid x_t)$ in the presence of local covariances – *i.e.* covariances in the neighborhood of x_t.

We illustrate this behavior in Figure 4.1 and measure its impact as the Sinkhorn divergence [72, 73] between the posteriors $p(x_t \mid y)$ and $q(x_t \mid y)$ when the prior $p(x)$ lies on randomly generated 1-dimensional manifolds [74] embedded in $\mathbb{R}^3$. The prior $p(x)$ is modeled as a mixture of isotropic Gaussians centered around points of the manifold, which gives access to $p(x_t)$, $\mathbb{E}[x \mid x_t]$ and $\mathbb{V}[x \mid x_t]$ analytically. The results, presented in Figure 4.2, indicate that using $\mathbb{V}[x \mid x_t]$ instead of heuristics leads to orders of magnitude more accurate posteriors $q(x_t \mid y)$. We expect this gap to further increase with real high-dimensional data as the latter often lies along low-dimensional manifolds and, therefore, presents strong local covariances.

Because the MCEM algorithm is sensitive to the accuracy of posterior samples [65–68], we choose to estimate $\mathbb{V}[x \mid x_t]$ using Tweedie's covariance formula [57–60]

$$\mathbb{V}[x \mid x_t] = \Sigma_t + \Sigma_t \nabla^2_{x_t} \log p(x_t)\, \Sigma_t \tag{4.21}$$

$$= \Sigma_t \nabla^\top_{x_t} \mathbb{E}[x \mid x_t] \approx \Sigma_t \nabla^\top_{x_t} d_\theta(x_t, t)\,. \tag{4.22}$$

Conjugate gradient method As noted by Finzi et al. [26], explicitly computing and materializing the Jacobian $\nabla^\top_{x_t} d_\theta(x_t, t) \in \mathbb{R}^{N \times N}$ is intractable in high dimension. Furthermore, even if we had access to $\mathbb{V}[x \mid x_t]$, naively computing the inverse of the matrix $\Sigma_y + A\mathbb{V}[x \mid x_t]A^\top$ in Eq. (4.20) would still be intractable. Fortunately, we observe that the matrix $\Sigma_y + A\mathbb{V}[x \mid x_t]A^\top$ is symmetric positive definite (SPD) and, therefore,

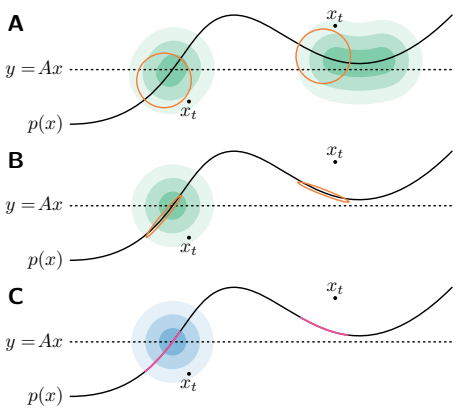


Figure 4.1. Illustration of the posterior $q(x_t \mid y)$ for the Gaussian approximation $q(x \mid x_t)$ when the prior $p(x)$ lies on a manifold. Ellipses represent 95 % credible regions of $q(x \mid x_t)$. (**A**) With Σ_t as heuristic for $\mathbb{V}[x \mid x_t]$, any x_t whose mean $\mathbb{E}[x \mid x_t]$ is close to the plane $y = Ax$ is considered likely. (**B**) With $\mathbb{V}[x \mid x_t]$, more regions are correctly pruned. (**C**) Ground-truth $p(x_t \mid y)$ and $p(x \mid x_t)$ for reference.

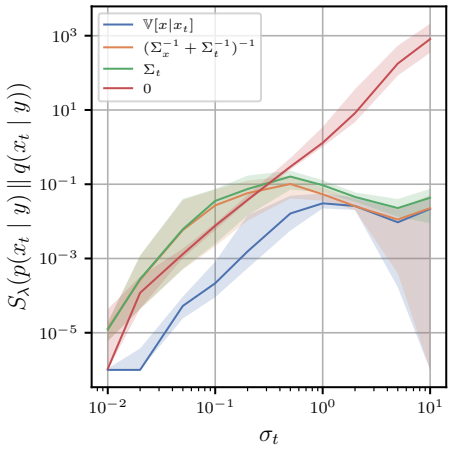


Figure 4.2. Sinkhorn divergence [72] between the posteriors $p(x_t \mid y)$ and $q(x_t \mid y)$ for different heuristics of $\mathbb{V}[x \mid x_t]$ when the prior $p(x)$ lies on 1-d manifolds embedded in $\mathbb{R}^3$. Lines and shades represent the 25-50-75 percentiles for 64 randomly generated manifolds [74] and measurement matrices $A \in \mathbb{R}^{1\times 3}$. Using $\mathbb{V}[x \mid x_t]$ instead of heuristics leads to orders of magnitude more accurate posteriors $q(x_t \mid y)$.

compatible with the conjugate gradient (CG) method [75]. The CG method is an iterative algorithm to solve linear systems of the form $Mv = b$ where the SPD matrix M and the vector b are known. Importantly, the CG method only requires implicit access to M through an operator that performs the matrix-vector product Mv given a vector v. In our case, the linear system to solve is

$$y - A\mathbb{E}[x \mid x_t] = \left(\Sigma_y + A\mathbb{V}[x \mid x_t]A^\top\right) v \tag{4.23}$$

$$= \Sigma_y v + A\Big(\underbrace{v^\top A\,\Sigma_t \nabla^\top_{x_t}\mathbb{E}[x \mid x_t]}_{\text{vector-Jacobian product}}\Big)^\top. \tag{4.24}$$

Within automatic differentiation frameworks [76, 77], the vector-Jacobian product in the right-hand side can be cheaply evaluated. In practice, due to numerical errors and imperfect training, the Jacobian $\nabla^\top_{x_t} d_\theta(x_t, t) \approx \nabla^\top_{x_t}\mathbb{E}[x \mid x_t]$ is not always perfectly SPD. Consequently, the CG method becomes unstable after a number of iterations and fails to reach an exact solution. Fortunately, we find that truncating the CG algorithm to very few iterations (1 to 3) already leads to significant improvements over using heuristics for the covariance $\mathbb{V}[x \mid x_t]$. Alternatively, the CG method can be replaced by other iterative algorithms that can solve non-symmetric non-definite linear systems, such as GMRES [78] or BiCGSTAB [79], at the cost of slower convergence.

4.5 Results

We conduct three experiments to demonstrate the effectiveness of DiEM. We design the first experiment around a low-dimensional latent variable x whose ground-truth distribution $p(x)$ is known. In this setting, we can use asymptotically exact sampling schemes such as predictor-corrector sampling [25, 48] or twisted diffusion sampling [80] without worrying about their computational cost. This allows us to validate our expectation-maximization pipeline (see Algorithm 5) in the limit of (almost) exact posterior sampling. The remaining experiments target two benchmarks from previous studies: corrupted CIFAR-10 and accelerated MRI. These tasks provide a good understanding of how our method would perform in typical empirical Bayes applications with limited data and compute.

4.5.1 Low-dimensional manifold

In this experiment, the latent variable $x \in \mathbb{R}^5 \sim p(x)$ lies on a random 1-dimensional manifold embedded in $\mathbb{R}^5$ represented in Figure 4.7. Each observation $y \in \mathbb{R}^2 \sim \mathcal{N}(y \mid Ax, \Sigma_y)$ is the result of a random linear projection of a latent x plus isotropic Gaussian noise ($\Sigma_y = 10^{-4} I$). The rows of the measurement matrix $A \in \mathbb{R}^{2\times 5}$ are drawn uniformly from the unit sphere $\mathbb{S}^4$. We note that observing all push-forward distributions $p(u^\top x)$ with $u \in \mathbb{S}^{N-1}$ of a distribution $p(x)$ in $\mathbb{R}^N$ is sufficient to recover $p(x)$ in theory [81, 82]. In practice, we generate a finite training set of 2^{16} pairs (y, A).

We train a DM $q_\theta(x)$ parameterized by a multi-layer perceptron $d_\theta(x_t, t)$ for $K = 32$ EM iterations. We apply Algorithm 7 to estimate the posterior score $\nabla_{x_t} \log q_\theta(x_t \mid y, A)$, but rely on the predictor-corrector [25, 48] sampling scheme with a large number (4096) of correction steps to sample from the posterior $q_\theta(x \mid y, A)$. We provide additional details such as noise schedule, network architectures, and learning rate in Appendix 4.C.

As expected, the model $q_{\theta_k}(x)$ converges towards a stationary distribution whose marginals are close to the marginals of the ground-truth $p(x)$, as illustrated in Figure 4.3. We attribute the remaining artifacts to finite data and inaccuracies in our sampling scheme.

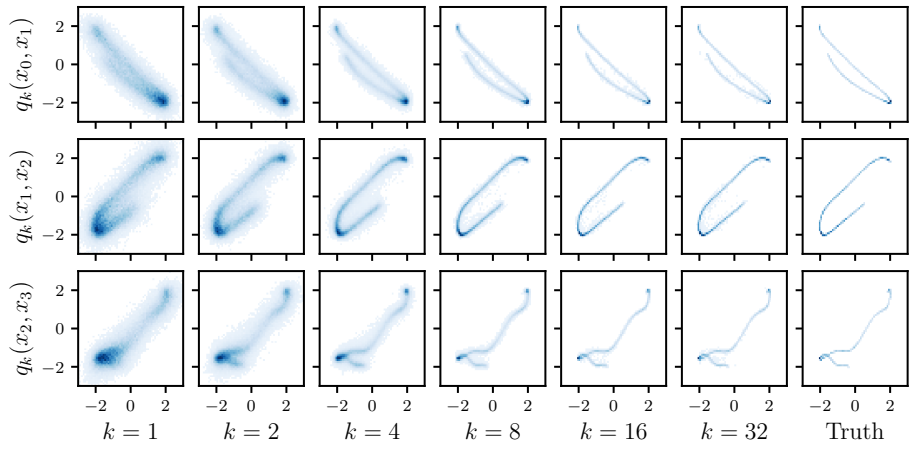


Figure 4.3. Illustration of 2-d marginals of the model $q_{\theta_k}(x)$ along the EM iterations. The initial Gaussian prior $q_0(x)$ leads to a very dispersed first model $q_{\theta_1}(x)$. The EM algorithm gradually prunes the density regions which are inconsistent with observations, until it reaches a stationary distribution. The marginals of the final distribution are close to the marginals of the ground-truth distribution.

Method	ρ	FID ↓	IS ↑
AmbientDiffusion [83]	0.20	11.70	7.97
	0.40	18.85	7.45
	0.60	28.88	6.88
DiEM w/ Tweedie	0.25	5.88	8.83
	0.50	6.76	8.75
	0.75	13.18	8.14
DiEM w/ $(I + \Sigma_t^{-1})^{-1}$	0.75	39.94	7.69
DiEM w/ Σ_t	0.75	118.69	4.23

Table 4.1. Evaluation of final models trained on corrupted CIFAR-10. DiEM outperforms AmbientDiffusion [83] at similar corruption ρ. Using heuristics for $\mathbb{V}[x \mid x_t]$ instead of Tweedie's formula greatly decreases the sample quality.

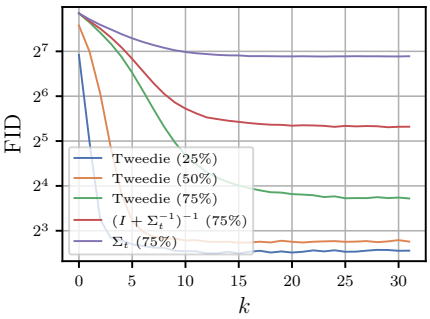


Figure 4.4. FID of $q_{\theta_k}(x)$ along the EM iterations for the corrupted CIFAR-10 experiment. With less corruption, the EM algorithm converges faster.

4.5.2 Corrupted CIFAR-10

Following Daras et al. [83], we take the 50 000 training images of the CIFAR-10 [84] dataset as latent variables x. A single observation y is generated for each image x by randomly deleting pixels with probability ρ. The measurement matrix A is therefore a binary diagonal matrix. We add negligible isotropic Gaussian noise ($\Sigma_y = 10^{-6} I$) for fair comparison with AmbientDiffusion [83] which cannot handle noisy observations.

For each corruption rate $\rho \in \{0.25, 0.5, 0.75\}$, we train a DM $q_\theta(x)$ parameterized by a U-Net [85] inspired network $d_\theta(x_t, t)$ for $K = 32$ EM iterations. We apply Algorithm 6 with $T = 256$ discretization steps and $\eta = 1$ to approximately sample from the posterior $q_\theta(x \mid y, A)$. We apply Algorithm 7 with several heuristics for $\mathbb{V}[x \mid x_t]$ to compare their results against Tweedie's covariance formula. For the latter, we truncate the conjugate gradient method in Algorithm 8 to a single iteration.

For each model $q_{\theta_k}(x)$, we generate a set of 50 000 images and evaluate its Inception score (IS) [86] and Fréchet Inception distance (FID) [87] against the uncorrupted training set of CIFAR-10. We report the results in Table 4.1 and Figures 4.4 and 4.5. At 75 % of corruption, our method performs similarly to AmbientDiffusion [83] at only 20 % of corruption. On the contrary, using heuristics for $\mathbb{V}[x \mid x_t]$ leads to poor sample quality.

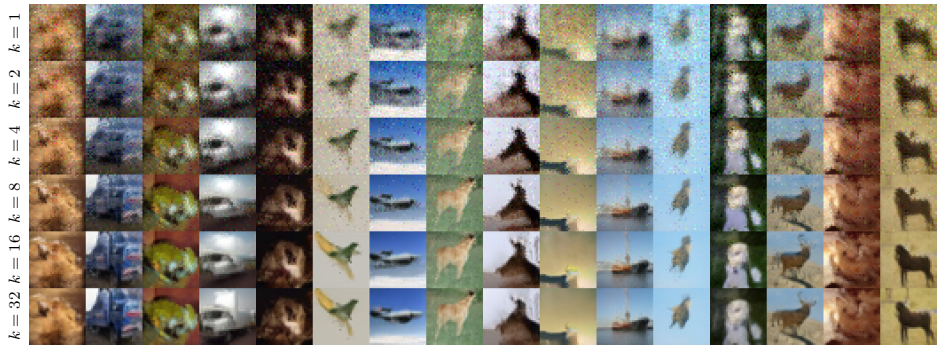


Figure 4.5. Example of samples from the model $q_{\theta_k}(x)$ along the EM iterations for the corrupted CIFAR-10 experiment with $\rho = 0.75$. We use the deterministic DDIM [52] sampling scheme for easier comparison. Generated images become gradually more detailed and less noisy.

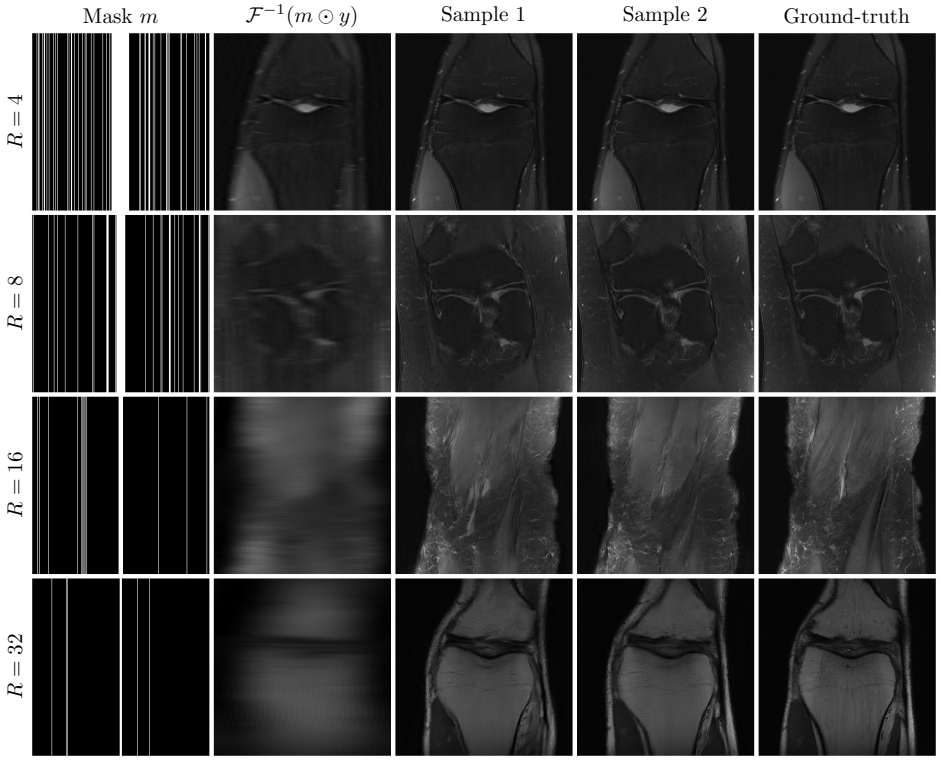


Figure 4.6. Examples of posterior samples for accelerated MRI using a diffusion prior trained from k-space observations only. Posterior samples are detailed and present plausible variations, while remaining consistent with the observation. We provide the zero-filled inverse $\mathcal{F}^{-1}(m \odot y)$ as baseline.

4.5.3 Accelerated MRI

Magnetic resonance imaging (MRI) is a non-invasive medical imaging technique used in radiology to inspect the internal anatomy and physiology of the body. MRI measurements of an object are obtained in the frequency domain, also called k-space, using strong magnetic fields. However, measuring the entire k-space can be time-consuming and expensive. Accelerated MRI [6–10] consists in inferring the scanned object based on partial, possibly randomized and noisy, frequency measurements.

In this experiment, following Kawar et al. [88], we consider the single-coil knee MRI scans from the fastMRI [9, 10] dataset. We treat each slice between the 10th and 40th of each scan as an independent latent variable x, represented as a 320×320 gray-scale image. Scans are min-max normalized such that pixel values range between -2 and 2. A single observation y is generated for each slice x by first applying the discrete Fourier transform $\mathcal{F}(x)$ and then a random horizontal frequency sub-sampling mask m with acceleration factor $R = 6$ [88, 89], meaning that a proportion $1/R$ of all frequencies are observed on average. Eventually, we obtain 24 853 k-space observations to which we add isotropic Gaussian noise ($\Sigma_y = 10^{-4} I$) to match Kawar et al. [88].

Once again, we train a DM $q_\theta(x)$ parameterized by a U-Net [85] inspired network $d_\theta(x_t, t)$ for $K = 16$ EM iterations. We apply Algorithm 6 with $T = 64$ discretization steps and $\eta = 1$ to approximately sample from the posterior $q_\theta(x \mid y, A)$ and truncate the conjugate gradient method in Algorithm 8 to 3 iterations. After training, we employ our final model $q_{\theta_K}(x)$ as a diffusion prior for accelerated MRI. Thanks to our moment matching posterior sampling, we are able to infer plausible scans at acceleration factors up to $R = 32$, as shown in Figure 4.6. Our samples are noticeably more detailed than the ones of Kawar et al. [88]. We choose not to report the PSNR/SSIM of our samples as these metrics only make sense for regression tasks and unfairly penalize proper generative models [90, 91]. We provide prior samples in Figure 4.13 and posterior samples for another kind of forward process in Figure 4.14.

4.6 Related Work

Empirical Bayes A number of previous studies have investigated the use of deep learning to solve the empirical Bayes problem. Louppe et al. [37] use adversarial training for learning a prior distribution that reproduces the empirical distribution of observations when pushed through a non-differentiable black-box forward process. Dockhorn et al. [35] use normalizing flows [92, 93] to estimate the prior density by variational inference when the forward process consists of additive noise. Vandegar et al. [38] also use normalizing flows but consider black-box forward processes for which the likelihood $p(y \mid x)$ is intractable. They note that empirical Bayes is an ill-posed problem in that distinct prior distributions may result in the same distribution over observations. Vetter et al. [39] address this issue by targeting the prior distribution of maximum entropy while minimizing the sliced-Wasserstein distance [81, 82] with the empirical distribution of observations. These methods rely on generative models $q_\theta(x)$ for which the density $q_\theta(x)$ or samples $x \sim q_\theta(x)$ are differentiable with respect to the parameters θ, which is not or hardly the case for diffusion models.

Closer to this work, Daras et al. [83] and Kawar et al. [88] attempt to train DMs from linear observations only. Daras et al. [83] consider noiseless observations of the form $y = Ax$ and train a network $d_\theta(Ax_t, A, t)$ to approximate $\mathbb{E}[x \mid Ax_t]$ under the assumption that $\mathbb{E}[A^\top A]$ is full-rank. The authors argue that $\mathbb{E}[x \mid Ax_t]$ can act as a surrogate for $\mathbb{E}[x \mid x_t]$. Similarly, Kawar et al. [88] assume Gaussian observations $y \sim \mathcal{N}(y \mid Ax, \Sigma_y)$ and train a network $d_\theta(Px_t, t)$ to approximate $\mathbb{E}[x \mid Px_t]$ under the assumption that $\mathbb{E}[P]$ is full-rank where $P = A^+A$ and A^+ is the Moore-Penrose pseudo-inverse of A.

The authors assume that $d_\theta(Px_t, t)$ can generalize to $P = I$, even if the training data does not contain $P = I$. In both cases, the trained networks are not proper denoisers approximating $\mathbb{E}[x \mid x_t]$ and cannot reliably parameterize a standard diffusion model, which is problematic for downstream applications. Notably, in the case of Bayesian inference, they require custom posterior sampling schemes such as the one proposed by Aali et al. [94] for AmbientDiffusion [83] models. Conversely, in this work, we do not make modifications to the denoising score matching objective [55, 56], which guarantees a proper DM that is compatible with any posterior sampling scheme at every iteration. In addition, we find that DiEM leads to quantitatively and qualitatively better diffusion priors.

In a concurrent work, Daras et al. [95] propose an algorithm to train DMs from noisy ($A = I$ and $\Sigma_y = \sigma_y^2 I$) data by enforcing the "consistency" of the denoiser along diffusion paths. They prove that the mean $\mathbb{E}[x \mid x_t]$ is the unique consistent denoiser. Interestingly, this training algorithm also relies on posterior samples, which are easy to obtain thanks to the white noise assumption.

Posterior sampling Recently, there has been much work on conditional generation using unconditional diffusion models, most of which adopt the posterior score decomposition in Eq. (4.16). As covered in Section 4.4.2, Chung et al. [23] propose an analytical approximation for the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ when the forward process $p(y \mid x)$ is Gaussian. Song et al. [24] and Rozet et al. [25] improve this approximation by taking the covariance $\mathbb{V}[x \mid x_t]$ into account in the form of simple heuristics. We build upon this idea and replace heuristics with a proper estimate of the covariance $\mathbb{V}[x \mid x_t]$ based on Tweedie's covariance formula [57–60]. Finzi et al. [26] take the same approach, but materialize the matrix $A\mathbb{V}[x \mid x_t]A^\top$ which is intractable in high dimension. Boys et al. [27] replace the covariance $\mathbb{V}[x \mid x_t]$ with a row-sum approximation $\mathrm{diag}(e^\top \mathbb{V}[x \mid x_t])$ where e is the all-ones vector. This approximation is only valid when $\mathbb{V}[x \mid x_t]$ is diagonal, which limits its applicability. Instead, we take advantage of the conjugate gradient method [75] to avoid materializing $A\mathbb{V}[x \mid x_t]A^\top$. A potential cheaper solution is to train a sparse approximation of $\mathbb{V}[x \mid x_t]$, as proposed by Peng et al. [96], but this approach is less general and not immediately applicable to any diffusion model.

A parallel line of work [97–99] extends the conditioning of diffusion models to arbitrary loss terms $\ell(x, y) \propto -\log p(y \mid x)$, for which no reliable analytical approximation of the likelihood score $\nabla_{x_t} \log p(y \mid x_t)$ exists. Song et al. [97] rely on Monte Carlo approximations of the likelihood $p(y \mid x_t) = \int p(y \mid x)\, p(x \mid x_t)\, \mathrm{d}x$ by sampling from a Gaussian approximation of $p(x \mid x_t)$. Conversely, He et al. [99] use the mean $\mathbb{E}[x \mid x_t]$ as a point estimate for $p(x \mid x_t)$, but leverage a pre-trained encoder-decoder pair to project the updates of x_t within its manifold. We note that our use of the covariance $\mathbb{V}[x \mid x_t]$ similarly leads to updates tangent to the manifold of x_t.

Finally, Wu et al. [80] propose a particle-based posterior sampling scheme that is asymptotically exact in the limit of infinitely many particles as long as the likelihood approximation $q(y \mid x_t)$ – here named the *twisting* function – converges to $p(y \mid x)$ as t approaches 0. Using TDS [80] as part of our expectation-maximization pipeline could lead to better results and/or faster convergence, at the cost of computational resources. In addition, the authors note that the efficiency of TDS [80] depends on how closely the twisting function approximates the exact likelihood. In this regard, our moment matching Gaussian approximation in Eq. (4.19) could be a good twisting candidate.

4.7 Discussion

To the best of our knowledge, we are the first to formalize the training of diffusion models from corrupted observations as an empirical Bayes [30–33] problem. In this work, we limit our analysis to linear Gaussian forward processes to take advantage of accurate and efficient high-dimensional posterior sampling schemes. This contrasts with typical empirical Bayes methods which target low-dimensional latent spaces and highly non-linear forward processes [35–39]. In addition, as mentioned in Section 4.6, these EB methods are not applicable to diffusion models. As such, we choose to benchmark DiEM against similar methods in the diffusion model literature [83, 88], but stress that a proper comparison with previous empirical Bayes methods would be valuable for both communities. We also note that Monte Carlo EM [61–68] can handle arbitrary forward processes $p(y \mid x)$ as long as one can sample from the posterior $q_\theta(x \mid y)$. Therefore, our approach could be adapted to any kind of forward processes in the future. We believe that the works of Dhariwal et al. [100] and Ho et al. [101] on diffusion guidance are good avenues for adapting our method to non-linear, non-differentiable, or even black-box forward processes.

From a computational perspective, the iterative nature of DiEM is a drawback compared to previous works [83, 88]. Notably, generating enough samples from the posterior can be expensive, although embarrassingly parallelizable. In addition, even though the architecture and training of the model $q_\theta(x)$ itself are simpler than in previous works [83, 88], the sampling step adds a significant amount of complexity, especially as the convergence of our method is sensitive to the quality of posterior samples. In fact, we find that previous posterior sampling methods [23–25, 27, 28] lead to disappointing results, which motivates us to develop a better one.

As such, moment matching posterior sampling (MMPS) is a byproduct of our work. However, it is not bound to the context of learning from observations and is applicable to any linear inverse problem given a pre-trained diffusion prior. In Appendix 4.E, we evaluate MMPS against previous posterior sampling methods for several linear inverse problems on the FFHQ [102] dataset. We find that MMPS consistently outperforms previous methods, both qualitatively and quantitatively. MMPS is remarkably stable and requires fewer sampling steps to generate qualitative samples, which largely makes up for its slightly higher step cost.

Finally, as mentioned in Section 4.6, empirical Bayes is an ill-posed problem in that distinct prior distributions may result in the same distribution over observations. In other words, it is generally impossible to identify "the" ground-truth distribution $p(x)$ given an empirical distribution of observations $p(y)$. Instead, for a sufficiently expressive diffusion model, our EM method will eventually converge to a prior $q_\theta(x)$ that is consistent with $p(y)$, but generally different from $p(x)$. Following the maximum entropy principle, as advocated by Vetter et al. [39], is left to future work.

Acknowledgments

François Rozet and Gérôme Andry are research fellows of the F.R.S.-FNRS (Belgium) and acknowledge its financial support.

The present research benefited from computational resources made available on Lucia, the Tier-1 supercomputer of the Walloon Region, infrastructure funded by the Walloon Region under the grant no. 1910247. The computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under the grant no. 2.5020.11 and by the Walloon Region.

MRI data used in the preparation of this article were obtained from the NYU fastMRI Initiative database [9, 10]. As such, NYU fastMRI investigators provided data but did not participate in analysis or writing of this report. A listing of NYU fastMRI investigators, subject to updates, can be found at https://fastmri.med.nyu.edu/. The primary goal of fastMRI is to test whether machine learning can aid in the reconstruction of medical images.

References

[1] Werner Heisenberg. "Physics and Philosophy: The Revolution in Modern Science". 1958.

[2] François Rozet et al. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[3] S. J. Warren and S. Dye. "Semilinear Gravitational Lens Inversion". In *The Astrophysical Journal* (2003).

[4] Warren R. Morningstar et al. "Data-driven Reconstruction of Gravitationally Lensed Galaxies Using Recurrent Inference Machines". In *The Astrophysical Journal* (2019).

[5] Siddharth Mishra-Sharma and Ge Yang. "Strong Lensing Source Reconstruction Using Continuous Neural Fields". 2022.

[6] Shanshan Wang et al. "Accelerating magnetic resonance imaging via deep learning". In *International Symposium on Biomedical Imaging*. 2016.

[7] Kerstin Hammernik et al. "Learning a variational network for reconstruction of accelerated MRI data". In *Magnetic Resonance in Medicine* (2018).

[8] Yoseo Han, Leonard Sunwoo, and Jong Chul Ye. "k-Space Deep Learning for Accelerated MRI". In *Transactions on Medical Imaging* (2020).

[9] Jure Zbontar et al. "fastMRI: An Open Dataset and Benchmarks for Accelerated MRI". 2018.

[10] Florian Knoll et al. "fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning". In *Radiology: Artificial Intelligence* (2020).

[11] G. Cowan. "A survey of unfolding methods for particle physics". In *Conf. Proc. C* (2002).

[12] Volker Blobel. "Unfolding Methods in Particle Physics". In *PHYSTAT*. CERN, 2011.

[13] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[14] Yannick Trémolet. "Accounting for an imperfect model in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).

[15] Thomas M. Hamill. "Ensemble-based atmospheric data assimilation". In *Predictability of Weather and Climate*. 2006.

[16] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[17] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[19] Yang Song et al. "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models". In *International Conference on Learning Representations*. 2022.

[20] Bahjat Kawar, Gregory Vaksman, and Michael Elad. "SNIPS: Solving Noisy Inverse Problems Stochastically". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[21] Bahjat Kawar et al. "Denoising Diffusion Restoration Models". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[22] Alexandre Adam et al. "Posterior samples of source galaxies in strong gravitational lenses with score-based priors". 2022.

[23] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In *International Conference on Learning Representations*. 2023.

[24] Jiaming Song et al. "Pseudoinverse-Guided Diffusion Models for Inverse Problems". In *International Conference on Learning Representations*. 2023.

[25] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[26] Marc Anton Finzi et al. "User-defined Event Sampling and Uncertainty Quantification in Diffusion Models for Physical Dynamical Systems". In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[27] Benjamin Boys et al. "Tweedie Moment Projected Diffusions For Inverse Problems". 2023.

[28] Yuanzhi Zhu et al. "Denoising Diffusion Models for Plug-and-Play Image Restoration". In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2023.

[29] Noe Dia et al. "Bayesian Imaging for Radio Interferometry with Score-Based Priors". 2023.

[30] Herbert E. Robbins. "An Empirical Bayes Approach to Statistics". In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. 1956.

[31] George Casella. "An Introduction to Empirical Bayes Data Analysis". In *The American Statistician* (1985).

[32] Bradley P. Carlin and Thomas A. Louis. "Empirical Bayes: Past, Present and Future". In *Journal of the American Statistical Association* (2000).

[33] Bradley Efron. "Two Modeling Strategies for Empirical Bayes Estimation". In *Statistical Science* (2014).

[34] G. D'Agostini. "A multidimensional unfolding method based on Bayes' theorem". In *Nuclear Instruments and Methods in Physics Research* 362.2 (1995).

[35] Tim Dockhorn et al. "Density Deconvolution with Normalizing Flows". 2020.

[36] Anders Andreassen et al. "OmniFold: A Method to Simultaneously Unfold All Observables". In *Physical Review Letters* (2020).

[37] Gilles Louppe, Joeri Hermans, and Kyle Cranmer. "Adversarial Variational Optimization of Non-Differentiable Simulators". In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. 2019.

[38] Maxime Vandegar et al. "Neural Empirical Bayes: Source Distribution Estimation and its Applications to Simulation-Based Inference". In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. 2021.

[39] Julius Vetter et al. "Sourcerer: Sample-based Maximum Entropy Source Distribution Estimation". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[40] Bradley Efron. "Empirical Bayes deconvolution estimates". In *Biometrika* (2016).

[41] Balasubramanian Narasimhan and Bradley Efron. "deconvolveR: A G-Modeling Program for Deconvolution and Empirical Bayes Estimation". In *Journal of Statistical Software* (2020).

[42] H. O. Hartley. "Maximum Likelihood Estimation from Incomplete Data". In *Biometrics* (1958).

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum Likelihood from Incomplete Data Via the EM Algorithm". In *Journal of the Royal Statistical Society* (1977).

[44] C. F. Jeff Wu. "On the Convergence Properties of the EM Algorithm". In *The Annals of Statistics* (1983).

[45] Geoffrey J McLachlan and Thriyambakam Krishnan. "The EM algorithm and extensions". John Wiley & Sons, 2007.

[46] Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. "Statistical guarantees for the EM algorithm: From population to sample-based analysis". In *The Annals of Statistics* (2017).

[47] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[48] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In *International Conference on Learning Representations*. 2021.

[49] Brian D. O. Anderson. "Reverse-time diffusion equation models". In *Stochastic Processes and their Applications* 12.3 (1982).

[50] Simo Särkkä and Arno Solin. "Applied Stochastic Differential Equations". Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[51] Qinsheng Zhang and Yongxin Chen. "Fast Sampling of Diffusion Models with Exponential Integrator". In *International Conference on Learning Representations*. 2023.

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In *International Conference on Learning Representations*. 2021.

[53] Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[54] Yaron Lipman et al. "Flow Matching for Generative Modeling". In 2023.

[55] Aapo Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". In *Journal of Machine Learning Research* (2005).

[56] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". In *Neural Computation* (2011).

[57] M. C. K. Tweedie. "Functions of a statistical variate with given means, with special reference to Laplacian distributions". In *Mathematical Proceedings of the Cambridge Philosophical Society* (1947).

[58] Bradley Efron. "Tweedie's Formula and Selection Bias". In *Journal of the American Statistical Association* (2011).

[59] Kwanyoung Kim and Jong Chul Ye. "Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[60] Chenlin Meng et al. “Estimating High Order Gradients of the Data Distribution by Denoising”. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[61] Greg C. G. Wei and Martin A. Tanner. “A Monte Carlo Implementation of the EM Algorithm and the Poor Man’s Data Augmentation Algorithms”. In *Journal of the American Statistical Association* (1990).

[62] Gilles Celeux and Jean Diebolt. “A stochastic approximation type EM algorithm for the mixture problem”. In *Stochastics and Stochastic Reports* (1992).

[63] Bernard Delyon, Marc Lavielle, and Eric Moulines. “Convergence of a stochastic approximation version of the EM algorithm”. In *The Annals of Statistics* (1999).

[64] James G. Booth and James P. Hobert. “Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm”. In *Journal of the Royal Statistical Society* (1999).

[65] Richard A. Levine and George Casella. “Implementations of the Monte Carlo EM Algorithm”. In *Journal of Computational and Graphical Statistics* 10.3 (2001).

[66] Brian S. Caffo, Wolfgang Jank, and Galin L. Jones. “Ascent-Based Monte Carlo Expectation-Maximization”. In *Journal of the Royal Statistical Society* (2005).

[67] Wolfgang Jank. “The EM Algorithm, Its Randomized Implementation and Global Optimization”. In *Perspectives in Operations Research*. 2006.

[68] William Ruth. “A review of Monte Carlo-based versions of the EM algorithm”. 2024.

[69] Michael E. Tipping and Christopher M. Bishop. “Mixtures of Probabilistic Principal Component Analyzers”. In *Neural Computation* (1999).

[70] Thomas Bayes. “An Essay Towards Solving a Problem in the Doctrine of Chances”. 1763.

[71] Christopher M. Bishop. “Pattern Recognition and Machine Learning”. Information Science and Statistics. Springer, 2006.

[72] Lénaïc Chizat et al. “Faster Wasserstein Distance Estimation with the Sinkhorn Divergence”. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[73] Rémi Flamary et al. “POT: Python Optimal Transport”. In *Journal of Machine Learning Research* (2021).

[74] Friedemann Zenke and Tim P. Vogels. “The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks”. In *Neural Computation* (2021).

[75] Magnus R. Hestenes and Eduard Stiefel. “Methods of Conjugate Gradients for Solving Linear Systems”. In *Journal of Research of the National Bureau of Standards* (1952).

[76] James Bradbury et al. “JAX: Composable transformations of Python + NumPy programs”. 2018.

[77] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[78] Youcef Saad and Martin Schultz. “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems”. In *Journal on Scientific and Statistical Computing* (1986).

[79] H. A. Van der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In *Journal on Scientific and Statistical Computing* (1992).

[80] Luhuan Wu et al. "Practical and Asymptotically Exact Conditional Sampling in Diffusion Models". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[81] Nicolas Bonneel et al. "Sliced and Radon Wasserstein Barycenters of Measures". In *Journal of Mathematical Imaging and Vision* (2015).

[82] Kimia Nadjahi et al. "Statistical and Topological Properties of Sliced Probability Divergences". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[83] Giannis Daras et al. "Ambient Diffusion: Learning Clean Distributions from Corrupted Data". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[84] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". Tech. rep. 2009.

[85] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In *Medical Image Computing and Computer-Assisted Intervention*. 2015.

[86] Tim Salimans et al. "Improved Techniques for Training GANs". In *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016.

[87] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[88] Bahjat Kawar et al. "GSURE-Based Diffusion Model Training with Corrupted Data". In *Transactions on Machine Learning Research* (2024).

[89] Ajil Jalal et al. "Robust Compressed Sensing MRI with Deep Generative Priors". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[90] Yochai Blau and Tomer Michaeli. "The Perception-Distortion Tradeoff". In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

[91] Mauricio Delbracio and Peyman Milanfar. "Inversion by Direct Iteration: An Alternative to Denoising Diffusion for Image Restoration". In *Transactions on Machine Learning Research* (2023).

[92] E. G. Tabak and Cristina V. Turner. "A Family of Nonparametric Density Estimation Algorithms". In *Communications on Pure and Applied Mathematics* 66.2 (2013).

[93] Danilo Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[94] Asad Aali et al. "Ambient Diffusion Posterior Sampling: Solving Inverse Problems with Diffusion Models Trained on Corrupted Data". In *International Conference on Learning Representations*. 2025.

[95] Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. "Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data". In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[96] Xinyu Peng et al. “Improving Diffusion Models for Inverse Problems Using Optimal Posterior Covariance”. 2024.

[97] Jiaming Song et al. “Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation”. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[98] Arpit Bansal et al. “Universal Guidance for Diffusion Models”. In *International Conference on Learning Representations*. 2024.

[99] Yutong He et al. “Manifold Preserving Guided Diffusion”. In *International Conference on Learning Representations*. 2024.

[100] Prafulla Dhariwal and Alexander Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[101] Jonathan Ho and Tim Salimans. “Classifier-Free Diffusion Guidance”. 2022.

[102] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[103] Ashish Vaswani et al. “Attention is All you Need”. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[104] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”. In *Neural Networks*. Special issue on deep reinforcement learning 107 (2018).

[105] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. “Layer Normalization”. 2016.

[106] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In *International Conference on Learning Representations*. 2015.

[107] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[108] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In *IEEE/CVF International Conference on Computer Vision*. 2023.

[109] Anton Obukhov et al. “High-fidelity performance metrics for generative models in PyTorch”. 2020.

[110] Wenzhe Shi et al. “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[111] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

[112] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In *Transactions on Image Processing* (2004).

4.A Algorithms

Algorithm 5 Expectation-maximization pipeline

Choose an initial prior $q_0(x)$
Initialize the parameters θ of the denoiser $d_\theta(x_t, t)$
for $k = 1$ to K **do**
    for $i = 1$ to S **do**
        $y_i, A_i \sim p(y, A)$
        $x_i \sim q_{k-1}(x \mid y_i, A_i)$ # Posterior sampling
    repeat
        $i \sim \mathcal{U}(\{1, \dots, S\})$
        $t \sim \mathcal{U}(0, 1)$
        $z \sim \mathcal{N}(0, I)$
        $x_t \leftarrow x_i + \sigma_t\, z$
        $\ell \leftarrow \lambda_t \left\| d_\theta(x_t, t) - x_i \right\|^2$ # Denoising score matching
        $\theta \leftarrow \textsc{GradientDescent}(\theta, \nabla_\theta \ell)$
    until convergence
    $\theta_k \leftarrow \theta$
    $q_k := q_{\theta_k}$
return θ_K

Algorithm 6 DDIM-style posterior sampling

$x_1 \sim \mathcal{N}(0, \Sigma_1)$
for $i = T$ to 1 **do**
    $s \leftarrow {}^{i-1}/_{T}$
    $t \leftarrow {}^{i}/_{T}$
    $\hat{x} \leftarrow x_t + \Sigma_t\, s_\theta(x_t \mid y, A)$ # Estimate $\mathbb{E}[x \mid x_t, y, A]$
    $z \sim \mathcal{N}(0, I)$
    $x_s \leftarrow \hat{x} + \sigma_s \sqrt{1 - \eta \left(1 - \frac{\sigma_s^2}{\sigma_t^2}\right)} \frac{x_t - \hat{x}}{\sigma_t} + \sigma_s \sqrt{\eta \left(1 - \frac{\sigma_s^2}{\sigma_t^2}\right)}\, z$
return x_0

Algorithm 7 Moment matching posterior score

function $s_\theta(x_t \mid y, A)$ # Estimate $\nabla_{x_t} \log q_\theta(x_t \mid y, A)$
  $\hat{x} \leftarrow d_\theta(x_t, t)$
  if Tweedie **then**
    $\Sigma_{x|x_t} \leftarrow \Sigma_t \nabla_{x_t} d_\theta(x_t, t)^\top$
  else
    $\Sigma_{x|x_t} \leftarrow \Sigma_t$ or $(I + \Sigma_t^{-1})^{-1}$ or $(\Sigma_x^{-1} + \Sigma_t^{-1})^{-1}$
  $u \leftarrow \left(\Sigma_y + A\Sigma_{x|x_t} A^\top\right)^{-1} (y - A\hat{x})$ # Solve with conjugate gradient method
  $s_{y|x} \leftarrow \nabla_{x_t} d_\theta(x_t, t)^\top A^\top u$ # Estimate $\nabla_{x_t} \log q_\theta(y \mid x_t, A)$
  $s_x \leftarrow \Sigma_t^{-1}(\hat{x} - x_t)$ # Estimate $\nabla_{x_t} \log q_\theta(x_t)$
  return $s_x + s_{y|x}$

Algorithm 8 Conjugate gradient method

function ConjugateGradient(A, b, x_0)
  $r_0 \leftarrow b - Ax_0$
  $p_0 \leftarrow r_0$
  for $i = 0$ to $N - 1$ **do**
    if $\|r_i\| \leq \epsilon$ **then**
      return x_i
    $\alpha_i \leftarrow \dfrac{r_i^\top r_i}{p_i^\top A p_i}$
    $x_{i+1} \leftarrow x_i + \alpha_i p_i$
    $r_{i+1} \leftarrow r_i - \alpha_i A p_i$
    $\beta_i \leftarrow \dfrac{r_{i+1}^\top r_{i+1}}{r_i^\top r_i}$
    $p_{i+1} \leftarrow r_{i+1} + \beta_i p_i$
  return x_N

4.B Tweedie's formulae

Theorem 1. *For any distribution $p(x)$ and $p(x_t \mid x) = \mathcal{N}(x_t \mid x, \Sigma_t)$, the first and second moments of the distribution $p(x \mid x_t)$ are linked to the score function $\nabla_{x_t} \log p(x_t)$ through*

$$\mathbb{E}[x \mid x_t] = x_t + \Sigma_t \nabla_{x_t} \log p(x_t) \tag{4.25}$$
$$\mathbb{V}[x \mid x_t] = \Sigma_t + \Sigma_t \nabla^2_{x_t} \log p(x_t) \Sigma_t \tag{4.26}$$

We provide proofs of Theorem 1 for completeness, even though it is a well known result [57–60].

Proof.

$$\begin{aligned}
\nabla_{x_t} \log p(x_t) &= \frac{1}{p(x_t)} \nabla_{x_t} p(x_t) \\
&= \frac{1}{p(x_t)} \int \nabla_{x_t} p(x, x_t) \, \mathrm{d}x \\
&= \frac{1}{p(x_t)} \int p(x, x_t) \nabla_{x_t} \log p(x, x_t) \, \mathrm{d}x \\
&= \int p(x \mid x_t) \nabla_{x_t} \log p(x_t \mid x) \, \mathrm{d}x \\
&= \int p(x \mid x_t) \Sigma_t^{-1} (x - x_t) \, \mathrm{d}x \\
&= \Sigma_t^{-1} \mathbb{E}[x \mid x_t] - \Sigma_t^{-1} x_t
\end{aligned}$$

□

Proof.

$$\begin{aligned}
\nabla^2_{x_t} \log p(x_t) &= \nabla_{x_t} \nabla^\top_{x_t} \log p(x_t) \\
&= \nabla_{x_t} \mathbb{E}[x \mid x_t]^\top \Sigma_t^{-1} - \Sigma_t^{-1} \\
&= \left(\int \nabla_{x_t} p(x \mid x_t) \, x^\top \, \mathrm{d}x \right) \Sigma_t^{-1} - \Sigma_t^{-1} \\
&= \left(\int p(x \mid x_t) \nabla_{x_t} \log \frac{p(x_t \mid x)}{p(x_t)} \, x^\top \, \mathrm{d}x \right) \Sigma_t^{-1} - \Sigma_t^{-1} \\
&= \left(\int p(x \mid x_t) \Sigma_t^{-1} \big(x - \mathbb{E}[x \mid x_t] \big) \, x^\top \, \mathrm{d}x \right) \Sigma_t^{-1} - \Sigma_t^{-1} \\
&= \Sigma_t^{-1} \Big(\mathbb{E}\left[x x^\top \mid x_t \right] - \mathbb{E}[x \mid x_t] \, \mathbb{E}[x \mid x_t]^\top \Big) \Sigma_t^{-1} - \Sigma_t^{-1} \\
&= \Sigma_t^{-1} \mathbb{V}[x \mid x_t] \, \Sigma_t^{-1} - \Sigma_t^{-1}
\end{aligned}$$

□

4.C Experiment details

All experiments are implemented within the JAX [76] automatic differentiation framework. The code for all experiments along instructions to obtain the data are made available at https://github.com/francois-rozet/diem.

Diffusion models As mentioned in Section 4.2, in this work, we adopt the variance-exploding SDE [47] and the denoiser parameterization [53]. Following Karras et al. [53], we precondition our denoiser $d_\theta(x_t, t)$ as

$$d_\theta(x_t, t) = \frac{1}{\sigma_t^2 + 1} x_t + \frac{\sigma_t}{\sqrt{\sigma_t^2 + 1}} h_\theta \left(\frac{x_t}{\sqrt{\sigma_t^2 + 1}}, \log \sigma_t \right) \tag{4.27}$$

where $h_\theta(x, \log \sigma)$ is an arbitrary noise-conditional network. The scalar $\log \sigma$ is embedded as a vector using a sinusoidal positional encoding [103]. In our experiments, we use an exponential noise schedule

$$\sigma_t = \exp\left((1 - t) \log 10^{-3} + t \log 10^2\right), \tag{4.28}$$

loss weights $\lambda_t = \frac{1}{\sigma_t^2} + 1$ and sample t from a Beta distribution $\mathcal{B}(\alpha = 3, \beta = 3)$ during training.

Low-dimensional manifold The noise-conditional network $h_\theta(x, \log \sigma)$ is a multi-layer perceptron with 3 hidden layers of 256 neurons and SiLU [104] activation functions. A layer normalization [105] function is inserted after each activation. The input of the network is the concatenation of x_t and the noise embedding vector. We train the network with Algorithm 5 for $K = 32$ EM iterations. Each iteration consists of 16 384 optimization steps of the Adam [106] optimizer. The optimizer and learning rate are re-initialized after each EM iteration. Other hyperparameters are provided in Table 4.2.

Table 4.2. Hyperparameters for the low-dimensional manifold experiment.

Architecture	MLP
Input shape	(5)
Hidden features	(256, 256, 256)
Activation	SiLU
Normalization	LayerNorm
Optimizer	Adam
Weight decay	0.0
Scheduler	linear
Initial learning rate	1×10^{-3}
Final learning rate	1×10^{-6}
Gradient norm clipping	1.0
Batch size	1024
Steps per EM iteration	16 384
EM iterations	32

We apply Algorithm 7 to estimate the posterior score $\nabla_{x_t} \log p(x_t \mid y, A)$ and truncate Algorithm 8 to 3 iterations. We rely on the predictor-corrector [25, 48] sampling scheme to sample from the posterior $q_\theta(x \mid y, A)$. Following Rozet et al. [25], the predictor is a deterministic DDIM [52] step and the corrector is a Langevin Monte Carlo step. We

perform 4096 prediction steps, each followed by 1 correction step. At each EM iteration, we generate a single latent x for each pair (y, A).

We generate smooth random manifolds according to a procedure described by Zenke et al. [74]. We evaluate the Sinkhorn divergences using the `POT` [73] package with an entropic regularization factor $\lambda = 1e - 3$.

Corrupted CIFAR-10 The noise-conditional network $h_\theta(x, \log \sigma)$ is a U-Net [85] with residual blocks [107], SiLU [104] activation functions and layer normalization [105]. Each residual block is modulated with respect to the noise σ_t in the style of diffusion transformers [108]. A multi-head self-attention block [103] is inserted after each residual block at the last level of the U-Net. We train the network with Algorithm 5 for $K = 32$ EM iterations. Each iteration consists of 256 epochs over the training set (50 000 images). To prevent overfitting, images are augmented with horizontal flips and hue shifts. The optimizer is re-initialized after each EM iteration. Other hyperparameters are provided in Table 4.3.

Table 4.3. Hyperparameters for the corrupted CIFAR-10 and accelerated MRI experiments.

Experiment	corrupted CIFAR-10	accelerated MRI
Architecture	U-Net	U-Net
Input shape	(32, 32, 3)	(80, 80, 16)
Residual blocks per level	(5, 5, 5)	(3, 3, 3, 3)
Channels per level	(128, 256, 384)	(128, 256, 384, 512)
Attention heads per level	(0, 4, 0)	(0, 0, 0, 4)
Kernel size	3	3
Activation	SiLU	SiLU
Normalization	LayerNorm	LayerNorm
Optimizer	Adam	Adam
Weight decay	0.0	0.0
Learning rate	2×10^{-4}	10^{-4}
Gradient norm clipping	1.0	1.0
EMA decay	0.9999	0.999
Dropout	0.1	0.1
Augmentation	h-flip, hue	h-flip, pad & crop
Batch size	256	256
Epochs per EM iteration	256	64
EM iterations	32	16

We apply Algorithm 6 with $T = 256$ discretization steps and $\eta = 1$ to sample from the posterior $q_\theta(x \mid y, A)$. We apply Algorithm 7 with several heuristics for $\mathbb{V}[x \mid x_t]$ to compare their results against Tweedie's covariance formula. For the latter, we truncate the conjugate gradient method in Algorithm 8 to a single iteration. At each EM iteration, we generate a single latent x for each pair (y, A). Each EM iteration (including sampling and training) takes around 4 h on 4 A100 (40GB) GPUs.

We evaluate the Inception score (IS) [86] and Fréchet Inception distance (FID) [87] of generated images using the `torch-fidelity` [109] package.

Accelerated MRI The noise-conditional network architecture is the same as for the corrupted CIFAR-10 experiment. The $320 \times 320 \times 1$ tensor x_t is reshaped into a $80 \times 80 \times 16$ tensor using pixel shuffling [110] before entering the network. We train the network

with Algorithm 5 for $K = 16$ EM iterations. Each iteration consists of 64 epochs over the training set ($2 \times 24\,853$ images). To prevent overfitting, images are augmented with horizontal flips and random crops. The optimizer is re-initialized after each EM iteration. Other hyperparameters are provided in Table 4.3.

We apply Algorithm 6 with $T = 64$ discretization steps and $\eta = 1$ to sample from the posterior $q_\theta(x \mid y, A)$. We truncate the conjugate gradient method in Algorithm 8 to 3 iterations. At each EM iteration, we generate 2 latents x for each pair (y, A), which acts as data augmentation. Each EM iteration (including sampling and training) takes around 3 h on 4 A100 (40GB) GPUs.

4.D Additional figures

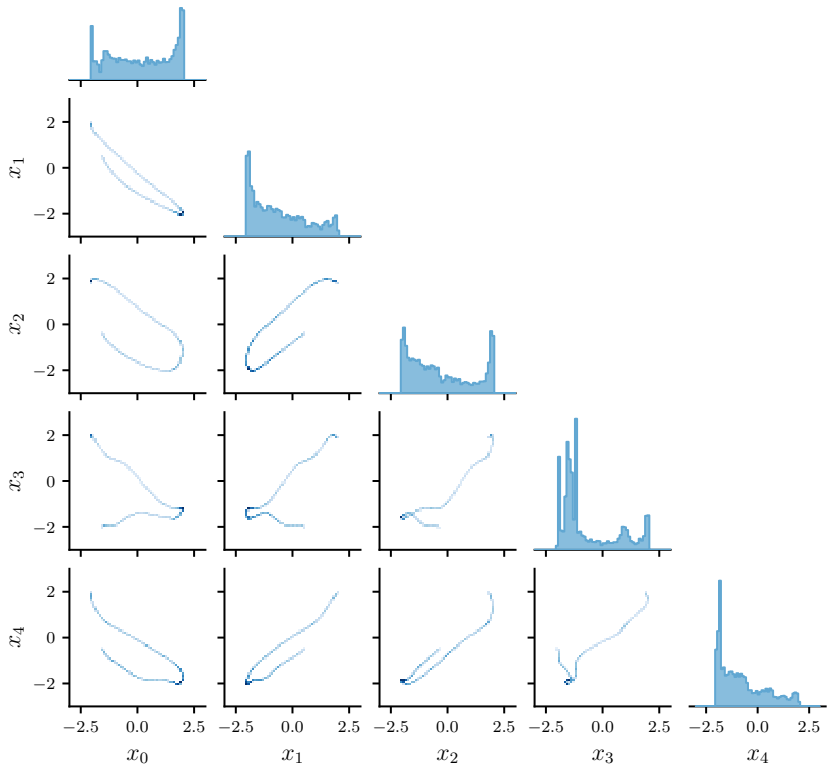


Figure 4.7. 1-d and 2-d marginals of the ground-truth distribution $p(x)$ used in the low-dimensional manifold experiment. The distribution lies on a random 1-dimensional manifold embedded in $\mathbb{R}^5$.

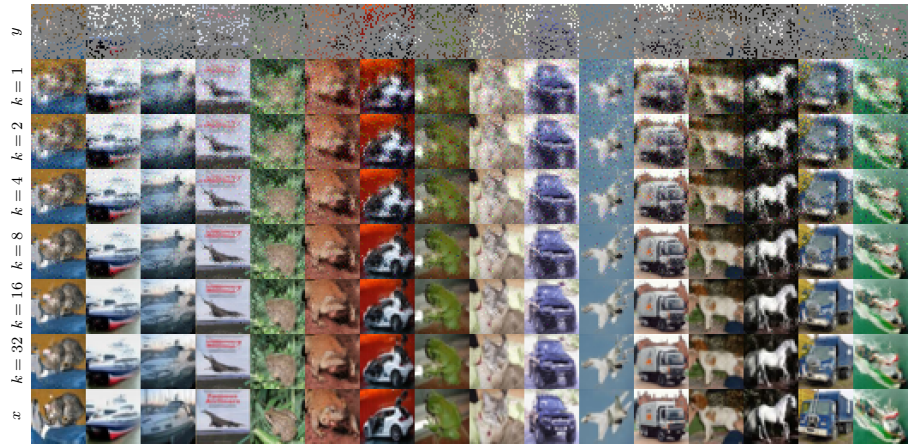


Figure 4.8. Example of samples from the posterior $q_{\theta_k}(x \mid y)$ along the EM iterations for the CIFAR-10 experiment. The generated images become gradually more detailed and less noisy.

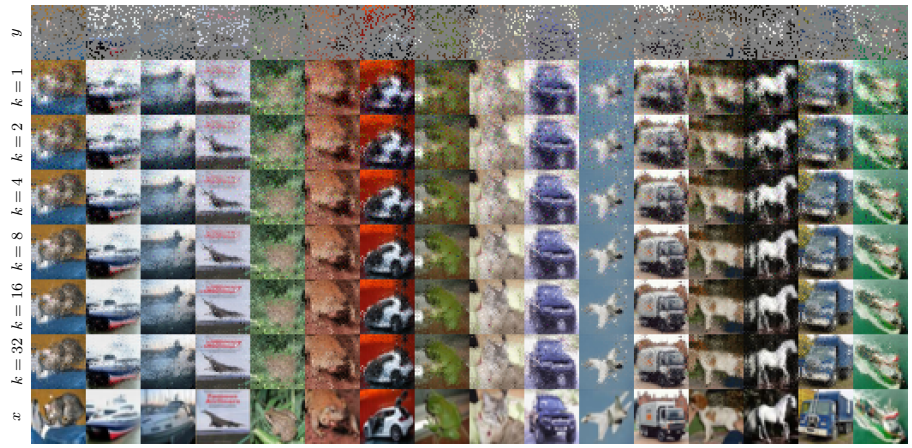


Figure 4.9. Example of samples from the posterior $q_{\theta_k}(x \mid y)$ along the EM iterations for the CIFAR-10 experiment when the heuristic $(I + \Sigma_t^{-1})^{-1}$ is used for $\mathbb{V}[x \mid x_t]$. The generated images become gradually more detailed but some noise remains.

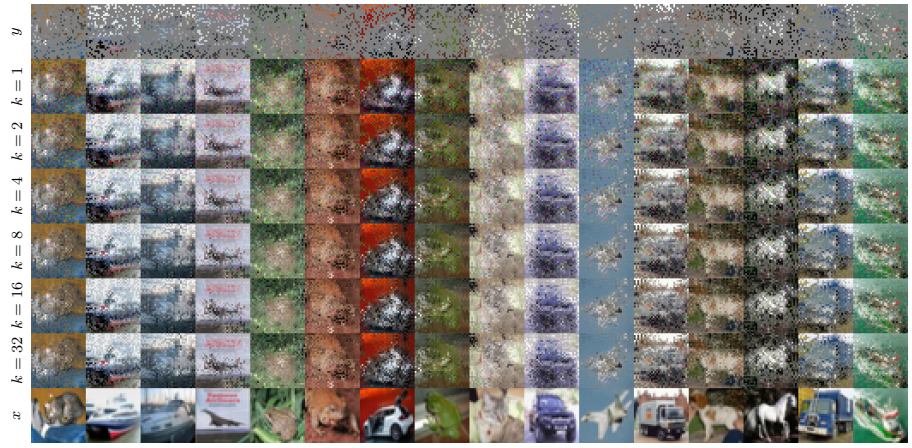


Figure 4.10. Example of samples from the posterior $q_{\theta_k}(x \mid y)$ along the EM iterations for the CIFAR-10 experiment when the heuristic Σ_t is used for $\mathbb{V}[x \mid x_t]$. The generated images remain very noisy.

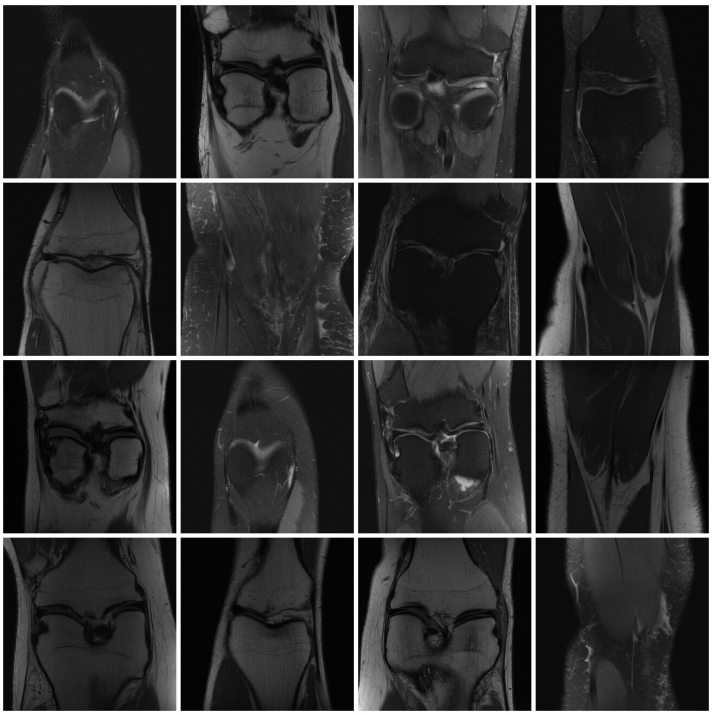


Figure 4.11. Example of scan slices from the fastMRI [9, 10] dataset.

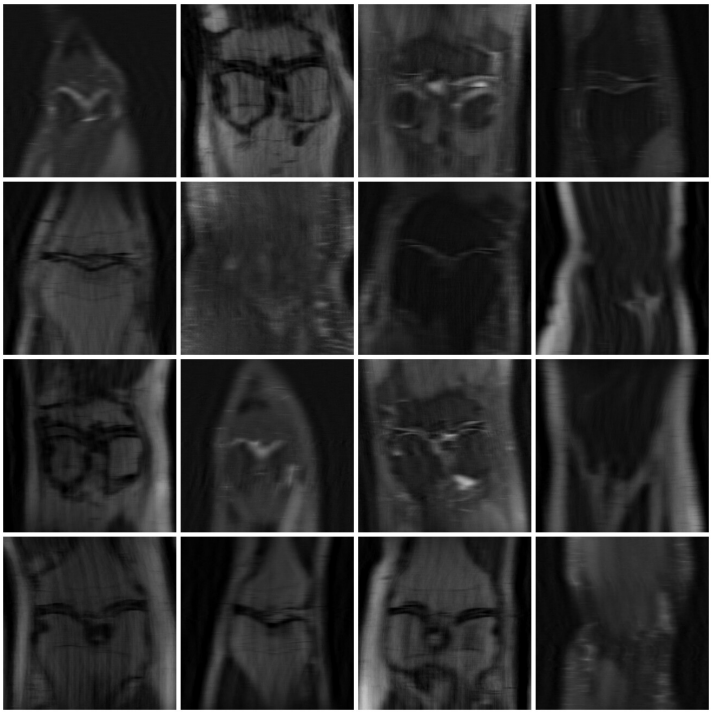


Figure 4.12. Example of k-space sub-sampling observations with acceleration factor $R = 6$ for the accelerated MRI experiment. We represent each observation by its zero-filled inverse, where missing frequencies are set to zero before taking the inverse discrete Fourier transform.

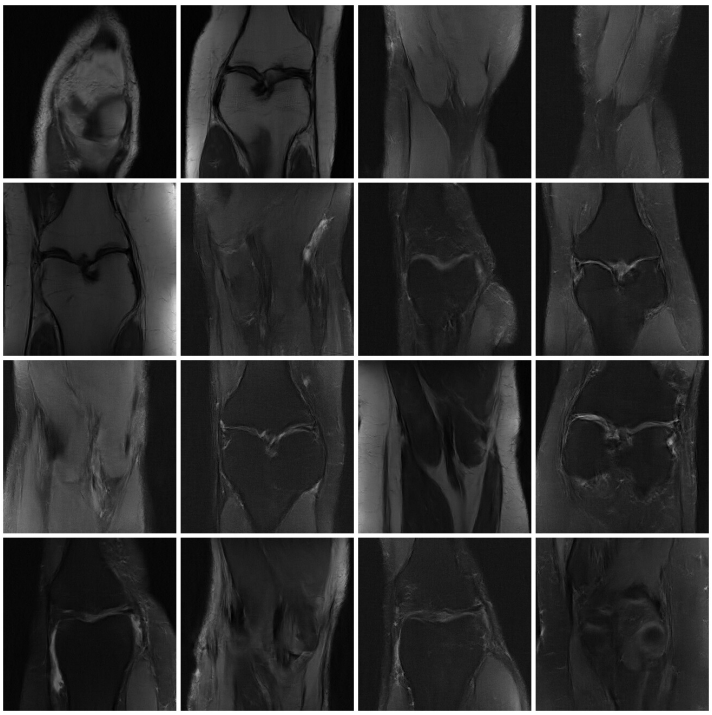


Figure 4.13. Example of samples from the final model $q_{\theta_k}(x)$ for the accelerated MRI experiment. The samples present varied and coherent global structures. Samples seem slightly less sharp than real scans (see Figure 4.11), but do not present artifacts typical to unresolved frequencies (see Figure 4.12).

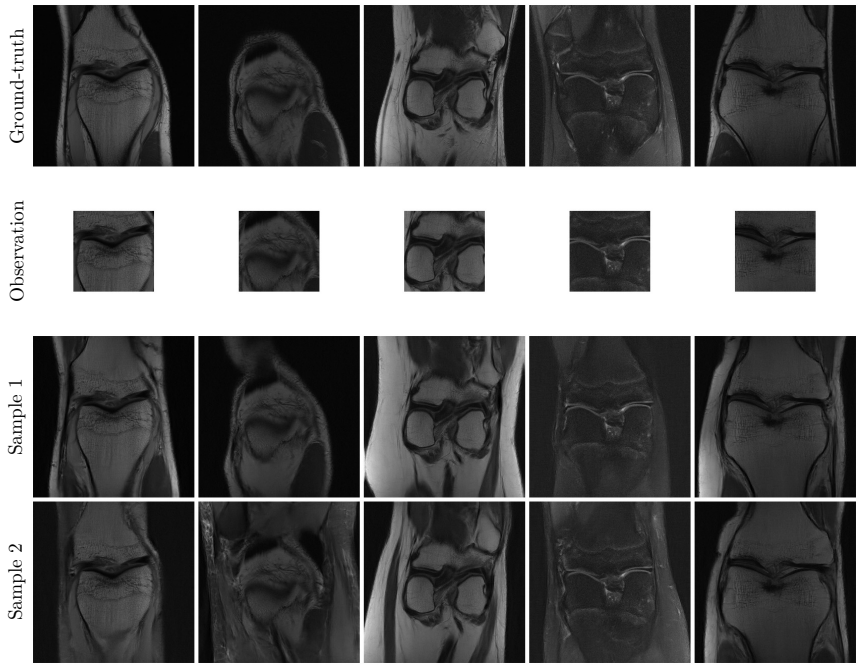


Figure 4.14. Examples of posterior samples using a diffusion prior trained from k-space observations only. The forward process crops the latent x to a centered 160×160 window. Moment matching posterior sampling is used to sample from the posterior. Samples are consistent with the ground-truth where observed, but present plausible variations elsewhere.

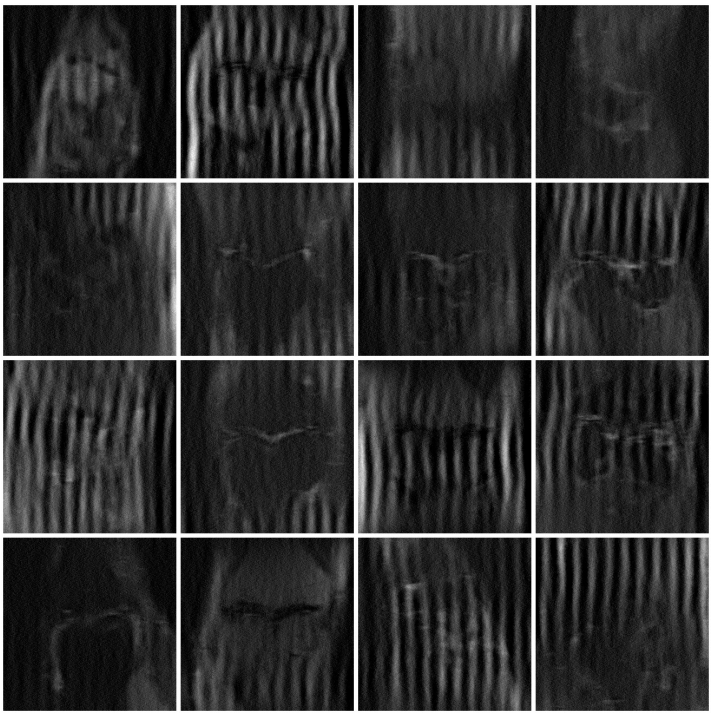


Figure 4.15. Example of samples from the model $q_{\theta_k}(x)$ after $k = 2$ EM iterations for the accelerated MRI experiment when the heuristic $(I + \Sigma_t^{-1})^{-1}$ is used for $\mathbb{V}[x \mid x_t]$. The samples start to present vertical artifacts due to poor sampling.

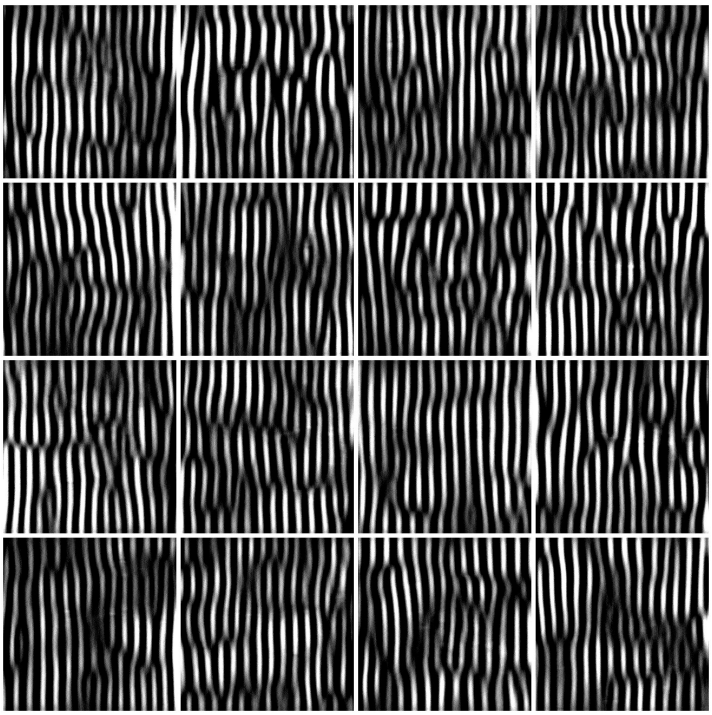


Figure 4.16. Example of samples from the model $q_{\theta_k}(x)$ after $k = 4$ EM iterations for the accelerated MRI experiment when the heuristic $(I + \Sigma_t^{-1})^{-1}$ is used for $\mathbb{V}[x \mid x_t]$. The artifacts introduced by the poor sampling get amplified at each iteration, leading to a total collapse after few iterations.

4.E Evaluation of MMPS

In this section, we evaluate moment matching posterior sampling (MMPS) presented in Section 4.4.2 independently from the context of learning from observations. The code is available at `https://github.com/francois-rozet/mmps-benchmark`.

Tasks We consider four linear inverse problems on the 256×256 FFHQ [102] dataset. (i) For box inpainting, we mask out a randomly positioned 128×128 square of pixels and add a large amount of noise ($\sigma_y = 1$). (ii) For random inpainting, we randomly delete pixels with 98 % probability and add a small amount of noise ($\sigma_y = 0.01$). (iii) For motion deblur, we apply a randomly generated 61×61 motion blur kernel and add a medium amount of noise ($\sigma_y = 0.1$). (iv) For super resolution, we apply a 4× bicubic downsampling and add a medium amount of noise ($\sigma_y = 0.1$).

Methods For all inverse problems, we use the pre-trained diffusion model provided by Chung et al. [23] as diffusion prior. We adapt and extend the DPS [23] codebase to support MMPS as well as DiffPIR [28], ΠGDM [24] and TMPD [27]. We use the DDIM [52] sampler with $\eta = 1$ for all methods, which is equivalent to the DDPM [18] sampler. We fine-tune the hyperparameters of DPS ($\zeta' = 0.5$) and DiffPIR ($\lambda = 8.0$) to have the best results across the four tasks. With MMPS, we find that the Jacobian of the pre-trained model provided by Chung et al. [23] is strongly non-symmetric and non-definite for large σ_t, which leads to unstable conjugate gradient (CG) [75] iterations. We therefore replace the CG solver with the GMRES [78] solver, which can solve non-symmetric non-definite linear systems.

Protocol We generate one observation per inverse problem for 100 images[1] of the FFHQ [102] dataset. We generate a sample for each observation with all considered posterior sampling methods. All methods are executed with the same random seed. We compute three standard image reconstruction metrics – LPIPS [111], PSNR and SSIM [112] – for each sample and report their average in Table 4.4. We present generated samples for each inverse problem in Figures 4.17, 4.18 and 4.19.

As a side note, we emphasize that reconstruction metrics do not necessarily reflect the accuracy of the inferred posterior distribution, which we eventually care about. For example, PSNR and SSIM [112] favor smooth predictions such as the mean $\mathbb{E}[x \mid y]$ over actual samples from the posterior $p(x \mid y)$. Conversely, LPIPS [111] favors predictions which are *perceptually* similar to the reference, even if they are distorted. In general, it is impossible to simultaneously optimize for all reconstruction metrics [90, 91].

Results MMPS consistently outperforms all baselines, both qualitatively and quantitatively. As expected, performing more solver iterations improves the sample quality, especially when the Gram matrix $AA^\top$ is strongly non-diagonal, which is the case for the motion deblur task. However, the improvement shows rapidly diminishing returns, as the difference between 1 and 3 iterations is much larger than between 3 and 5. MMPS is also remarkably stable with respect to the number of sampling steps in contrast to DPS [23], DiffPIR [28] and ΠGDM [24] which are sensitive to the number of steps and choice of hyperparameters. Finally, MMPS requires fewer sampling steps to reach the same image quality as previous methods, which largely makes up for its slightly higher step cost.

[1] Chung et al. [23] do not indicate which subset of FFHQ [102] was used to train their model. Without further information, we choose to use the first 100 images for evaluation, which could lead to biased metrics if the diffusion prior was trained on them. However, since we use the same diffusion prior for all posterior sampling methods, the evaluation remains fair.

Table 4.4. Quantitative evaluation of MMPS with 1, 3 and 5 solver iterations.

		Box inpainting			**Random inpainting**			**Motion deblur**			**Super resolution**		
Method	Steps	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
DiffPIR [28]	10	0.33	19.17	0.50	0.78	10.97	0.32	0.24	24.54	0.72	0.20	26.63	0.78
DiffPIR [28]	100	0.30	18.15	0.54	0.68	10.26	0.25	0.19	23.97	0.70	0.17	25.24	0.73
DiffPIR [28]	1000	0.33	17.39	0.49	0.74	9.51	0.21	0.17	23.55	0.67	0.15	24.72	0.70
DPS [23]	10	0.64	10.41	0.34	0.58	12.68	0.43	0.75	8.63	0.27	0.58	12.01	0.41
DPS [23]	100	0.38	16.82	0.50	0.39	16.66	0.49	0.29	19.75	0.57	0.35	18.29	0.54
DPS [23]	1000	0.22	21.01	0.64	0.19	21.90	0.66	0.18	22.91	0.66	0.16	25.02	0.72
ΠGDM [24]	10	0.40	18.94	0.61	0.59	11.28	0.40	0.25	25.83	0.76	0.25	26.42	0.77
ΠGDM [24]	100	0.44	18.23	0.47	0.39	17.03	0.48	0.25	22.37	0.61	0.15	25.63	0.71
ΠGDM [24]	1000	0.81	14.80	0.31	**0.14**	22.32	0.69	1.06	13.12	0.21	0.64	18.41	0.29
TMPD [27]	10	0.36	19.90	0.64	0.59	11.08	0.40	0.27	25.28	0.74	0.26	26.07	0.76
TMPD [27]	100	0.27	19.86	0.64	0.58	10.73	0.31	0.17	26.22	0.76	0.17	26.79	0.77
TMPD [27]	1000	0.25	19.53	0.62	0.68	9.98	0.25	0.14	25.91	0.74	0.14	26.53	0.76
MMPS (1)	10	0.27	21.19	0.68	0.26	22.41	0.69	0.33	22.12	0.66	0.24	26.94	0.78
MMPS (1)	100	0.20	21.19	0.67	0.18	22.18	0.69	0.20	23.92	0.71	0.15	27.32	0.79
MMPS (1)	1000	**0.19**	20.77	0.64	0.18	21.94	0.66	0.16	23.83	0.69	0.12	26.92	0.77
MMPS (3)	10	0.26	21.55	0.68	0.21	23.58	0.74	0.24	25.33	0.75	0.19	27.94	**0.81**
MMPS (3)	100	0.20	21.29	0.67	0.15	22.76	0.71	0.15	26.16	0.76	0.13	27.18	0.78
MMPS (3)	1000	**0.19**	21.01	0.64	0.15	22.45	0.68	0.12	25.73	0.74	**0.11**	26.69	0.76
MMPS (5)	10	0.23	**21.73**	**0.69**	0.20	**23.72**	**0.75**	0.20	**26.70**	**0.78**	0.18	**28.02**	**0.81**
MMPS (5)	100	0.20	21.30	0.67	0.15	22.82	0.72	0.13	**26.70**	0.77	0.13	27.12	0.78
MMPS (5)	1000	0.20	20.98	0.64	**0.14**	22.52	0.69	**0.11**	26.18	0.75	**0.11**	26.60	0.76

Table 4.5. Time and memory complexity of MMPS for the 4× super resolution task. Each solver iteration increases the time per step by around 16 ms. The maximum memory allocated by MMPS is about 10 % larger than DPS [23] and ΠGDM [24].

Method	VJPs	Time [ms/step]	Memory [GB]
DiffPIR [28]	0	30.2	0.66
DPS [23]	1	40.5	2.29
ΠGDM [24]	1	47.6	2.30
TMPD [27]	2	62.2	2.52
MMPS (1)	2	58.0	2.52
MMPS (3)	4	90.1	2.52
MMPS (5)	6	122.1	2.52

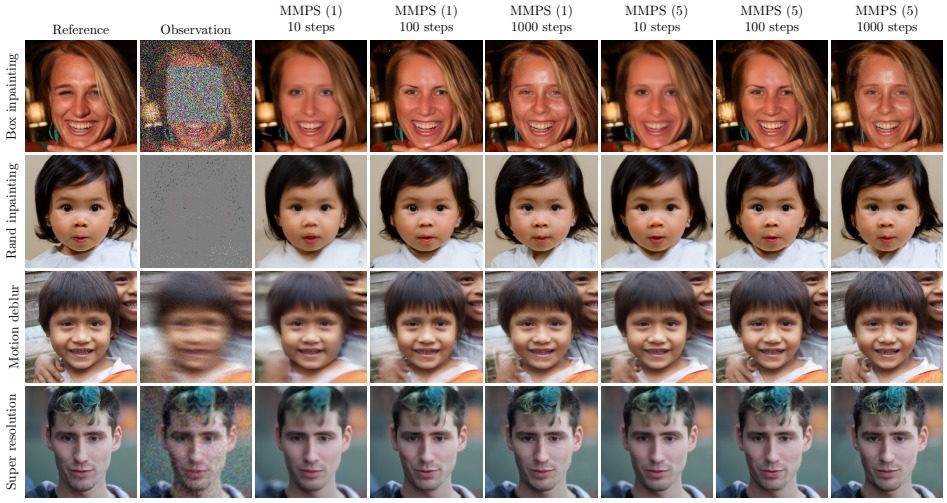


Figure 4.17. Qualitative evaluation of MMPS with 1 and 5 solver iterations.

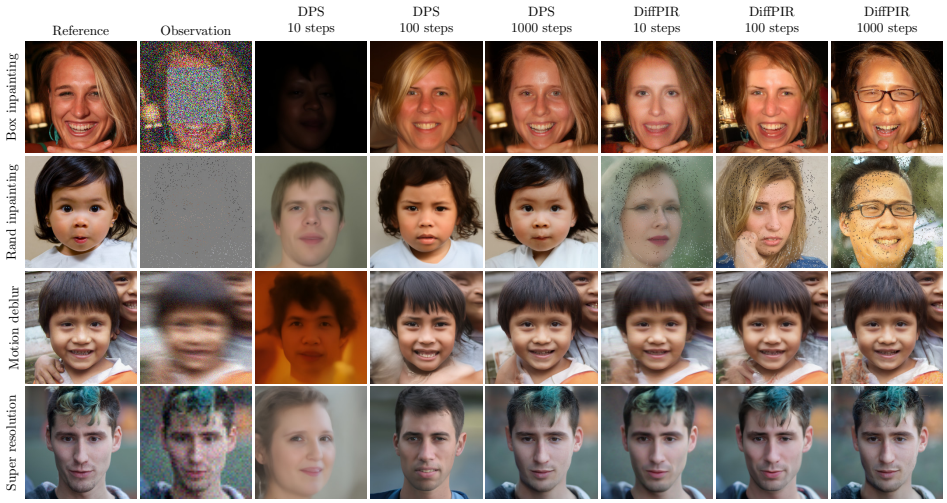


Figure 4.18. Qualitative evaluation of DPS [23] and DiffPIR [28].

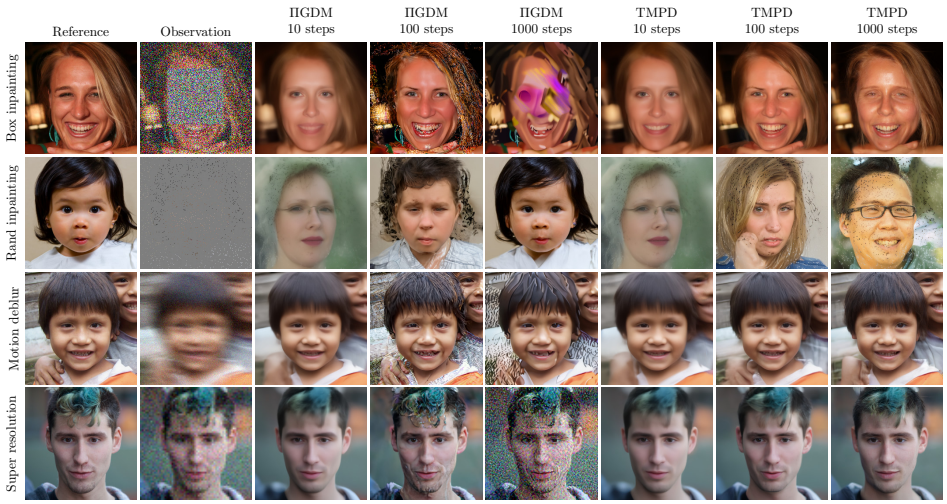


Figure 4.19. Qualitative evaluation of ΠGDM [24] and TMPD [27].

5 Lost in Latent Space

> *Data becomes temporarily interesting to some self-improving, but computationally limited, subjective observer once they learn to predict or compress the data in a better way, thus making it subjectively simpler and more beautiful.*
>
> — Jürgen Schmidhuber (2009)

Addendum

This chapter has previously been published as

> 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". In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

This project was conducted as part of an internship with Polymathic at the Flatiron Institute. The initial goal of the project, training latent diffusion models on physics data, was decided by François R., François L., Ruben, and Michael. As the leading author, François R. conducted the experiments, interpreted the results, and wrote the manuscript. Ruben, Michael, and Gilles offered guidance, suggested experiments, and participated in the writing and literature review. François L. and Shirley provided supervision and funding.

For the version presented in this chapter, we have slightly modified the placement of figures with respect to the original publication. The content remains otherwise unchanged.

Abstract

The steep computational cost of diffusion models at inference hinders their use as fast physics emulators. In the context of image and video generation, this computational drawback has been addressed by generating in the latent space of an autoencoder instead of the pixel space. In this work, we investigate whether a similar strategy can be effectively applied to the emulation of dynamical systems and at what cost. We find that the accuracy of latent-space emulation is surprisingly robust to a wide range of compression rates (up to 1000×). We also show that diffusion-based emulators are consistently more accurate than non-generative counterparts and compensate for uncertainty in their predictions with greater diversity. Finally, we cover practical design choices, spanning from architectures to optimizers, that we found critical to train latent-space emulators.

5.1 Introduction

Numerical simulations of dynamical systems are at the core of many scientific and engineering disciplines. Solving partial differential equations (PDEs) that describe the dynamics of physical phenomena enables, among others, weather forecasts [3, 4], predictions of solar wind and flares [5–7], or control of plasma in fusion reactors [8, 9]. These simulations typically operate on fine-grained spatial and temporal grids and require significant computational resources for high-fidelity results.

To address this limitation, a promising strategy is to develop neural network-based emulators to make predictions orders of magnitude faster than traditional numerical solvers. The typical approach [10–19] is to consider the dynamics as a function $f(x^i) = x^{i+1}$ that evolves the state x^i of the system and to train a neural network $f_\phi(x)$ to approximate that function. In the context of PDEs, this network is sometimes called a neural solver [13, 20, 21]. After training, the autoregressive application of the solver, or rollout, emulates the dynamics. However, recent studies [13, 20–23] reveal that, while neural solvers demonstrate impressive accuracy for short-term prediction, errors accumulate over the course of the rollout, leading to distribution shifts between training and inference. This phenomenon is even more severe for stochastic or undetermined systems, where it is not possible to predict the next state given the previous one(s) with certainty. Instead of modeling the uncertainty, neural solvers produce a single point estimate, usually the mean, instead of a distribution.

The natural choice to alleviate these issues are generative models, in particular diffusion models, which have shown remarkable results in recent years. Following their success, diffusion models have been applied to emulation tasks [20, 21, 24–27] for which they were found to mitigate the rollout instability of non-generative emulators. However, diffusion models are much more expensive than deterministic alternatives at inference, due to their iterative sampling process, which defeats the purpose of using an emulator. To address this computational drawback, many works in the image and video generation literature [28–34] consider generating in the latent space of an autoencoder. This approach has been adapted with success to the problem of emulating dynamical systems [35–39], sometimes even outperforming pixel-space emulation. In this work, we seek to answer a simple question: *What is the impact of latent-space compression on emulation accuracy?* To this end, we train and systematically evaluate latent-space emulators across a wide range of compression rates for challenging dynamical systems from TheWell [40]. Our results indicate that

i. Latent diffusion-based emulation is surprisingly robust to the compression rate, even when autoencoder reconstruction quality greatly degrades.

ii. Latent-space emulators match or exceed the accuracy of pixel-space emulators, while using fewer parameters and less training compute.

iii. Diffusion-based emulators consistently outperform their non-generative counterparts in both accuracy and plausibility of the emulated dynamics.

Finally, we dedicate part of this manuscript to design choices. We discuss architectural and modeling decisions for autoencoders and diffusion models that enable stable training of latent-space emulators under high compression. To encourage further research in this direction, we provide the code for all experiments along pre-trained model weights at `https://github.com/polymathicai/lola`.

5.2 Diffusion models

The primary purpose of diffusion models (DMs) [41, 42], also known as score-based generative models [43, 44], is to generate plausible data from a distribution $p(x)$ of interest. Formally, continuous-time diffusion models define a series of increasingly noisy distributions

$$p(x_t) = \int p(x_t \mid x)\, p(x)\, \mathrm{d}x = \int \mathcal{N}(x_t \mid \alpha_t\, x, \sigma_t^2 I)\, p(x)\, \mathrm{d}x \tag{5.1}$$

such that the ratio $\alpha_t/\sigma_t \in \mathbb{R}_+$ is monotonically decreasing with the time $t \in [0, 1]$. For such a series, there exists a family of reverse-time stochastic differential equations (SDEs) [44–46]

$$\mathrm{d}x_t = \left[f_t\, x_t - \frac{1+\eta^2}{2} g_t^2\, \nabla_{x_t} \log p(x_t) \right] \mathrm{d}t + \eta\, g_t\, \mathrm{d}w_t \tag{5.2}$$

where $\eta \geq 0$ is a parameter controlling stochasticity, the coefficients f_t and g_t are derived from α_t and σ_t [44–46], and for which the variable x_t follows $p(x_t)$. In other words, we can draw noise samples $x_1 \sim p(x_1) \approx \mathcal{N}(0, \sigma_1^2 I)$ and obtain data samples $x_0 \sim p(x_0) \approx p(x)$ by solving Eq. (5.2) from $t = 1$ to 0. For high-dimensional samples, the terminal signal-to-noise ratio α_1/σ_1 should be at or very close to zero [47]. In this work, we adopt the rectified flow [30, 48, 49] noise schedule, for which $\alpha_t = 1 - t$ and $\sigma_t = t$.

Denoising score matching In practice, the score function $\nabla_{x_t} \log p(x_t)$ in Eq. (5.2) is unknown, but can be approximated by a neural network trained via denoising score matching [50, 51]. Several equivalent parameterizations and objectives have been proposed for this task [42–44, 49, 52, 53]. In this work, we adopt the denoiser parameterization $d_\phi(x_t, t)$ and its objective [53]

$$\arg\min_\phi \mathbb{E}_{p(x)p(t)p(x_t|x)} \left[\lambda_t \left\| d_\phi(x_t, t) - x \right\|_2^2 \right], \tag{5.3}$$

for which the optimal denoiser is the mean $\mathbb{E}[x \mid x_t]$ of $p(x \mid x_t)$. Importantly, $\mathbb{E}[x \mid x_t]$ is linked to the score function through Tweedie's formula [54–57]

$$\mathbb{E}[x \mid x_t] = \frac{x_t + \sigma_t^2 \nabla_{x_t} \log p(x_t)}{\alpha_t}, \tag{5.4}$$

which allows to use $s_\phi(x_t) = \sigma_t^{-2}(d_\phi(x_t, t) - \alpha_t\, x_t)$ as a score estimate in Eq. (5.2).

5.3 Methodology

In this section, we detail and motivate our experimental methodology for investigating the impact of compression on the accuracy of latent-space emulators. To summarize, we consider three challenging datasets from TheWell [40]. For each dataset, we first train a series of autoencoders with varying compression rates. These autoencoders learn to map high-dimensional physical states $x^i \in \mathbb{R}^{H \times W \times C_{\text{pixel}}}$ to low-dimensional latent representations $z^i \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times C_{\text{latent}}}$. Subsequently, for each autoencoder, we train two emulators operating in the latent space: a diffusion model (generative) and a neural solver (non-generative). Both are trained to predict the next n latent states $z^{i+1:i+n}$ given the current latent state z^i and simulation parameters θ. This technique, known as temporal bundling [13], mitigates the accumulation of errors during rollout by decreasing the number of required autoregressive steps. After training, latent-space emulators are used to produce autoregressive rollouts $z^{1:L}$ starting from known initial state $z^0 = E_\psi(x^0)$ and simulation parameters θ, which are then decoded to the pixel space as $\hat{x}^i = D_\psi(z^i)$.

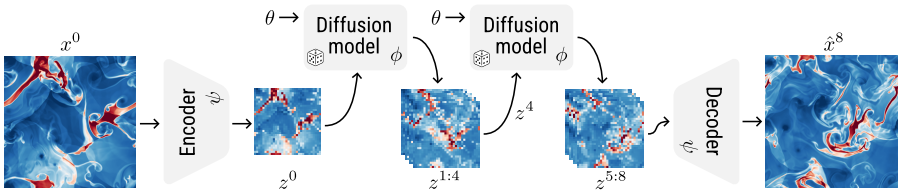


Figure 5.1. Illustration of the latent-space emulation process. At each step of the autoregressive rollout, the diffusion model generates the next $n = 4$ latent states $z^{i+1:i+n}$ given the current state z^i and the simulation parameters θ. After rollout, the generated latent states are decoded to pixel space.

5.3.1 Datasets

To study the effects of extreme compression rates, the datasets we consider should be high-dimensional and contain large amounts of data. Intuitively, the effective size of the dataset decreases in latent space, making overfitting more likely at fixed model capacity. According to these criteria, we select three datasets from TheWell [40]. Additional details are provided in Appendix 5.B.

Euler Multi-Quadrants The Euler equations model the behavior of compressible non-viscous fluids. In this dataset, the initial state presents multiple discontinuities which result in interacting shock waves as the system evolves for 100 steps. The 2d state of the system is represented with three scalar fields (energy, density, pressure) and one vector field (momentum) discretized on a 512×512 grid, for a total of $C_{\text{pixel}} = 5$ channels. Each simulation has either periodic or open boundary conditions and a different heat capacity γ, which constitutes their parameters θ. We set a time stride $\Delta = 4$ between consecutive states x^i and x^{i+1}, such that the simulation time $\tau = i \times \Delta$.

Rayleigh-Bénard (RB) The Rayleigh-Bénard convection phenomenon occurs when a horizontal layer of fluid is heated from below and cooled from above. Over the 200 simulation steps, the temperature difference leads to the formation of convection currents where cooler fluid sinks and warmer fluid rises. The 2d state of the system is represented with two scalar fields (buoyancy, pressure) and one vector field (velocity) discretized on a 512×128 grid, for a total of $C_{\text{pixel}} = 4$ channels. Each simulation has different Rayleigh and Prandtl numbers as parameters θ. We set a time stride $\Delta = 1$.

Turbulence Gravity Cooling (TGC) The interstellar medium can be modeled as a turbulent fluid subject to gravity and radiative cooling. Starting from a homogeneous state, dense filaments form in the fluid, leading to the birth of stars. The 3d state of the system is represented with three scalar fields (density, pressure, temperature) and one vector field (velocity) discretized on a $64 \times 64 \times 64$ grid, for a total of $C_{\text{pixel}} = 6$ channels. Each simulation has different initial conditions function of their density, temperature, and metallicity. We set a time stride $\Delta = 1$.

5.3.2 Autoencoders

To isolate the effect of compression, we use a consistent autoencoder architecture and training setup across datasets and compression rates. We focus on compressing individual states x^i into latent states $z^i = E_\psi(x^i)$, which are reconstructed as $\hat{x}^i = D_\psi(z^i)$.

Architecture We adopt a convolution-based autoencoder architecture similar to the one used by Rombach et al. [28], which we adapt to perform well under high compression

rates. Specifically, inspired by Chen et al. [33], we initialize the downsampling and upsampling layers near identity, which enables training deeper architectures with complex latent representations, while preserving reconstruction quality. For 2d datasets (Euler and RB), we set the spatial downsampling factor $r = 32$ for all autoencoders, meaning that a 32×32 patch in pixel space corresponds to one token in latent space. For 3d datasets (TGC), we set $r = 8$. The compression rate is then controlled solely by varying the number of channels per token in the latent representation. For instance, with the Euler dataset, an autoencoder with $C_{\text{latent}} = 64$ latent channels – `f32c64` in the notations of Chen et al. [33] – transforms the input state with shape $512 \times 512 \times 5$ to a latent state with shape $16 \times 16 \times 64$, yielding a compression rate of 80. This setup ensures that the architectural capacity remains similar for all autoencoders and allows for fair comparison across compression rates. Further details as well as a short ablation study are provided in Appendix 5.B.

Training Latent diffusion models [28] often rely on a Kullback-Leibler (KL) divergence penalty to encourage latents to follow a standard Gaussian distribution. However, this term is typically down-weighted by several orders of magnitude to prevent severe reconstruction degradation. As such, the KL penalty acts more as a weak regularization than a proper variational objective [58] and post-hoc standardization of latents is often necessary. We replace this KL penalty with a deterministic saturating function

$$z \mapsto \frac{z}{\sqrt{1 + z^2/B^2}} \tag{5.5}$$

applied to the encoder's output. In our experiments, we choose the bound $B = 5$ to mimic the range of a standard Gaussian distribution. We find this approach simpler and more effective at structuring the latent space, without introducing a tradeoff between regularization and reconstruction quality. We additionally omit perceptual [59] and adversarial [60, 61] loss terms, as they are designed for natural images where human perception is the primary target, unlike physics. The training objective thus simplifies to a reconstruction loss

$$\arg\min_{\psi} \mathbb{E}_{p(x)} \left[\ell(x, D_\psi(E_\psi(x))) \right] . \tag{5.6}$$

The loss ℓ is typically a variation of L_1 or L_2 regression, which we discuss in Appendix 5.B. Finally, we find that preconditioned optimizers [62–64] greatly accelerate the training convergence of autoencoders compared to the widespread Adam [65] optimizer (see Table 5.4). We adopt the PSGD [62] implementation in the `heavyball` [66] library for its fewer number of tunable hyper-parameters and lower memory footprint than SOAP [64].

5.3.3 Diffusion models

We train diffusion models to predict the next n latent states $z^{i+1:i+n}$ given the current state z^i and simulation parameters θ, that is to generate from $p(z^{i+1:i+n} \mid z^i, \theta)$. We parameterize our diffusion models with a denoiser $d_\phi(z_t^{i:i+n}, b, \theta, t)$ whose task is to denoise sequences of noisy states $z_t^i \sim p(z_t^i \mid z^i) = \mathcal{N}(z_t^i \mid \alpha_t\, z^i, \sigma_t^2 I)$ given the parameters θ of the simulation. Conditioning with respect to known elements in the sequence $z^{i:i+n}$ is tackled with a binary mask $b \in \{0, 1\}^{n+1}$ concatenated to the input, as in MCVD [67]. For instance, $b = (1, 0, \dots, 0)$ indicates that the first element z^i is known, while $b = (1, \dots, 1, 0)$ indicates that the first $n - 1$ elements $z^{i:i+n-1}$ are known. Known elements are provided to the denoiser without noise.

Architecture Drawing inspiration from recent successes in latent image generation [29–33], we use a transformer-based architecture for the denoiser. We incorporate

several architectural refinements shown to improve performance and stability, including query-key normalization [68], rotary positional embedding (RoPE) [69, 70], and value residual learning [71]. The transformer operates on the spatial and temporal axes of the input $z_t^{i:i+n}$, while the parameters θ and diffusion time t modulate the transformer blocks. Thanks to the considerable ($r = 32$) spatial downsampling performed by the autoencoder, we are able to apply full spatio-temporal attention, avoiding the need for sparse attention patterns [72–74]. Finally, we fix the token embedding size (1024) and the number of transformer blocks (16) for all diffusion models. The only architectural variation stems from the number of input and output channels dictated by the corresponding autoencoder.

Training As in Section 5.2, diffusion models are trained via denoising score matching [50, 51]

$$\arg\min_{\phi} \mathbb{E}_{p(\theta, z^{i:i+n}, z_t^{i:i+n}) p(b)} \left[\left\| d_\phi(\underbrace{z^{i:i+n} \odot b}_{\text{clean}} + \underbrace{z_t^{i:i+n} \odot (1-b)}_{\text{noisy}}, b, \theta, t) - z^{i:i+n} \right\|_2^2 \right] \tag{5.7}$$

with the exception that the data does not come from the pixel-space distribution $p(\theta, x^{1:L})$ but from the latent-space distribution $p(\theta, z^{1:L})$ determined by the encoder E_ψ. Following Voleti et al. [67], we randomly sample the binary mask $b \sim p(b)$ during training to cover several conditioning tasks, including prediction with variable-length context $p(z^{i+c:i+n} \mid z^{i:i+c-1})$.

Sampling After training, we sample from the learned distribution by solving Eq. (5.2) with $\eta = 0$, which corresponds to the probability flow ODE [44]. To this end, we implement a 3rd order Adams-Bashforth multi-step integration method, as proposed by Zhang et al. [75]. Intuitively, this method leverages information from previous integration steps to improve accuracy. We find this approach highly effective, producing high-quality samples with significantly fewer neural function evaluations (NFEs) than other widespread samplers [52, 53].

5.3.4 Neural solvers

We train neural solvers to perform the same task as diffusion models. Unlike the latter, however, solvers do not generate from $p(z^{i+1:i+n} \mid z^i, \theta)$, but produce a point estimate $f_\phi(z_i, \theta) \approx \mathbb{E}\left[z^{i+1:i+n} \mid z_i, \theta\right]$ instead. We also train a pixel-space neural solver, for which $z^i = x^i$, as baseline.

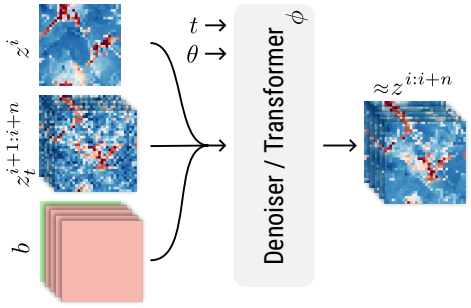


Figure 5.2. Illustration of the denoiser's inputs and outputs, while generating from $p(z^{i+1:i+n} \mid z^i, \theta)$. A binary mask b is concatenated to the input to indicate which elements are noisy. The transformer blocks are modulated with respect to the parameters θ and the diffusion time t.

Architecture For latent-space neural solvers, we use the same transformer-based architecture as for diffusion models. The only notable difference is that transformer blocks are only modulated with respect to the simulation parameters θ. For the pixel-space neural solver, we keep the same architecture, but group the pixels into 16×16 patches, as in vision transformers [76]. We also double the token embedding size (2048) such that the pixel-space neural solver has roughly two times more trainable parameters than an autoencoder and latent-space emulator combined.

Training Neural solvers are trained via mean regression

$$\arg\min_{\phi} \mathbb{E}_{p(\theta, z^{i:i+n}) p(b)} \left[\left\| f_\phi(z^{i:i+n} \odot b, b, \theta) - z^{i:i+n} \right\|_2^2 \right] . \tag{5.8}$$

Apart from the training objective, the training configuration (optimizer, learning rate schedule, batch size, epochs, masking, ...) for neural solvers is strictly the same as for diffusion models.

5.3.5 Evaluation metrics

We consider several metrics for evaluation, each serving a different purpose. We report these metrics either at a lead time $\tau = i \times \Delta$ or averaged over a lead time horizon $a : b$. If the states x^i present several fields, the metric is first computed on each field separately, then averaged.

Variance-normalized RMSE The root mean squared error (RMSE) and its normalized variants are widespread metrics to quantify the point-wise accuracy of an emulation [23, 40, 77]. Following Ohana et al. [40], we pick the variance-normalized RMSE (VRMSE) over the more common normalized RMSE (NRMSE), as the latter down-weights errors in non-negative fields such as pressure and density. Formally, for two spatial fields u and v, the VRMSE is defined as

$$\mathrm{VRMSE}(u, v) = \sqrt{\frac{\langle (u - v)^2 \rangle}{\langle (u - \langle u \rangle)^2 \rangle + \epsilon}} \tag{5.9}$$

where $\langle \cdot \rangle$ denotes the spatial mean operator and $\epsilon = 10^{-6}$ is a numerical stability term.

Power spectrum RMSE For chaotic systems such as turbulent fluids, it is typically intractable to achieve accurate long-term emulation as very small errors can lead to entirely different trajectories later on. In this case, instead of reproducing the exact trajectory, emulators should generate diverse trajectories that remain statistically plausible. Intuitively, even though structures are wrongly located, the types of patterns and their distribution should stay similar [78]. Following Ohana et al. [40], we assess statistical plausibility by comparing the power spectra of the ground-truth and emulated trajectories. For two spatial fields u and v, we compute the isotropic power spectra p_u and p_v and split them into three frequency bands (low, mid and high) evenly distributed in log-space. We report the RMSE of the relative power spectra p_v / p_u over each band.

Spread-skill ratio In earth sciences [27, 77], the skill of an ensemble of K particles is defined as the RMSE of the ensemble mean. The spread is defined as the ensemble standard deviation. Under these definitions and the assumption of a perfect forecast where ensemble particles are exchangeable, Fortin et al. [77] show that

$$\text{Skill} \approx \sqrt{K+1 / K}\ \text{Spread} . \tag{5.10}$$

This motivates the use of the (corrected) spread-skill ratio as a metric. Intuitively, if the ratio is smaller than one, the ensemble is biased or under-dispersed. If the ratio is larger than one, the ensemble is over-dispersed. It should be noted, however, that a spread-skill ratio of 1 is a necessary but insufficient condition for a perfect forecast.

5.4 Results

We start with the evaluation of the autoencoders. For all datasets, we train three autoencoders with respectively 64, 16, and 4 latent channels. These correspond to compression rates of 80, 320 and 1280 for the Euler dataset, 64, 256, and 1024 for the RB dataset, and 48, 192, 768 for the TGC dataset, respectively. In the following, we refer to models by their compression rate. Additional experimental details are provided in Section 5.3 and Appendix 5.B.

For each autoencoder, we evaluate the reconstruction $\hat{x}^i = D_\psi(E_\psi(x^i))$ of all states x^i in 64 test trajectories $x^{0:L}$. As expected, when the compression rate increases, the reconstruction quality degrades, as reported in Figure 5.3. For the Euler dataset, the reconstruction error grows with the lead time due to wavefront interactions and rising high-frequency content. For the RB dataset, the reconstruction error peaks mid-simulation during the transition from low to high-turbulence regime. Similar trends can be observed for the power spectrum RMSE in Tables 5.8, 5.9 and 5.10, where the high-frequency band is most affected by compression. These results so far align with what practitioners intuitively expect from lossy compression.

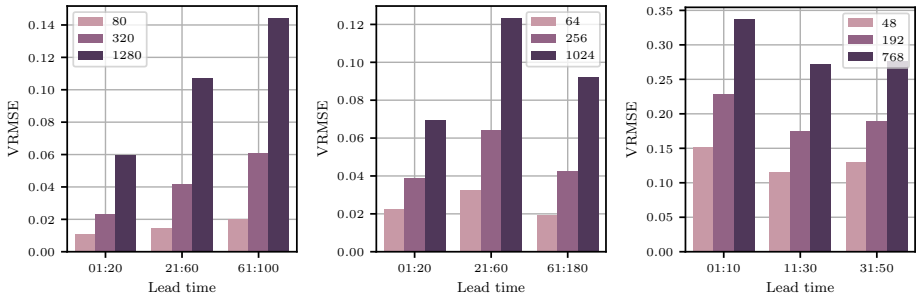


Figure 5.3. Average VRMSE of the autoencoder reconstruction at different compression rates and lead time horizons for the Euler (left), RB (center) and TGC (right) datasets. The compression rate has a clear impact on reconstruction quality.

We now turn to the evaluation of the emulators. For each autoencoder, we train two latent-space emulators: a diffusion model and a neural solver. Starting from the initial state $z^0 = E_\psi(x^0)$ and simulation parameters θ of 64 test trajectories $x^{0:L}$, each emulator produces 16 distinct autoregressive rollouts $z^{1:L}$, which are then decoded to the pixel space as $\hat{x}^i = D_\psi(z^i)$. Note that for neural solvers, all 16 rollouts are identical. We compute the metrics of each prediction $\hat{x}^i$ against the ground-truth state x^i.

As expected from imperfect emulators, the emulation error grows with the lead time, as shown in Figures 5.5 and 5.8. However, the point-wise error of diffusion models, as measured by the VRMSE, does not grow (Euler, TGC) and sometimes decreases (RB) with higher compression rates. Even at extreme (> 1000) compression rates, latent-space emulators outperform the baseline pixel-space neural solver, despite the latter benefiting from more parameters and training compute. Similar observations can be made with the power spectrum RMSE over low and mid-frequency bands. High-frequency content, however, appears limited by the autoencoder's reconstruction capabilities. We confirm this hypothesis by recomputing the metrics relative to the autoencoded state $D_\psi(E_\psi(x^i))$,

which we report in Figure 5.9. This time, the power spectrum RMSE of the diffusion models is low for mid and high-frequency bands. These findings support a puzzling narrative: emulation accuracy exhibits strong resilience to latent-space compression, starkly contrasting with the clear degradation in reconstruction quality.

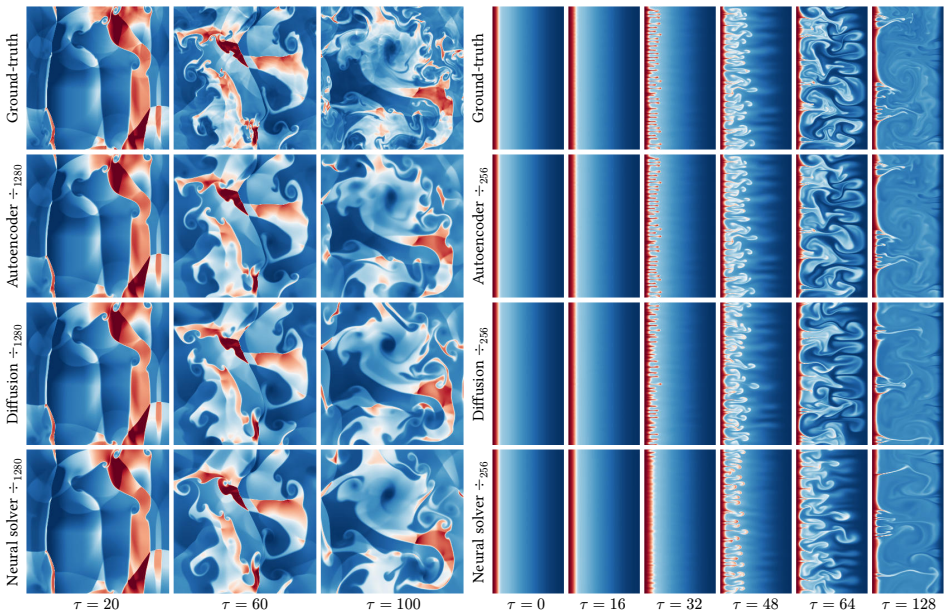


Figure 5.4. Examples of latent-space emulation for the Euler (left) and Rayleigh-Bénard (right) datasets. Even for large compression rates (÷), latent-space emulators are able to reproduce the dynamics surprisingly faithfully, despite significant reconstruction artifacts. For Euler, wavefronts are accurately propagated until the end of the simulation, while vortices are well located, but distorted. For Rayleigh-Bénard, diffusion-based emulators produce plumes that grow at the correct pace but diverge from the ground-truth. Similar observations can be made in Figures 5.10 to 5.21.

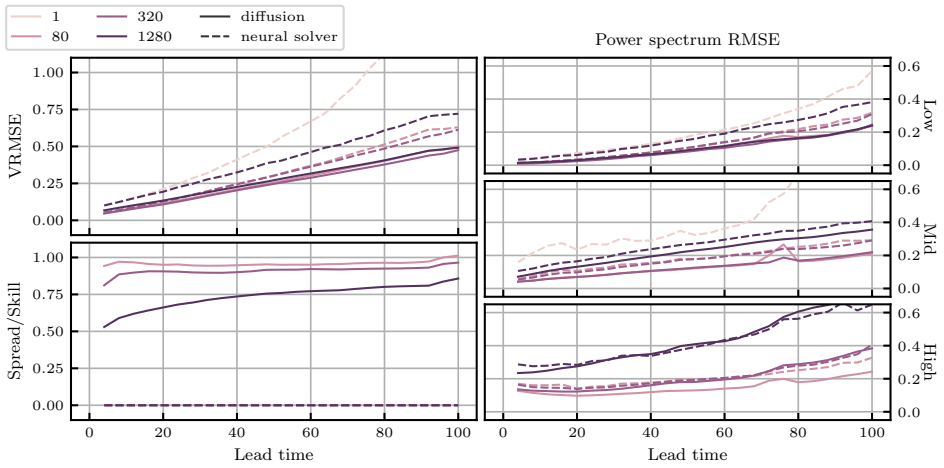


Figure 5.5. Average evaluation metrics of latent-space emulation for the Euler dataset. As expected from imperfect emulators, the emulation error grows with the lead time. However, the compression rate has little to no impact on diffusion-based emulation accuracy, beside high-frequency content. The spread-skill ratio [27, 77] drops slightly with the compression rate, which could be a sign of overfitting. Diffusion-based emulators are consistently more accurate than neural solvers.

Our experiments also provide a direct comparison between generative (diffusion) and deterministic (neural solver) approaches to emulation within a latent space. Figures 5.8 and 5.9 indicate that diffusion-based emulators are consistently more accurate than their deterministic counterparts and generate trajectories that are statistically more plausible in terms of power spectrum. This can be observed qualitatively in Figure 5.4 or Figures 5.10 to 5.21 in Appendix 5.C. In addition, the spread-skill ratio of diffusion models is close to 1, suggesting that the ensemble of trajectories they produce are reasonably well calibrated in terms of uncertainty. However, the ratio slightly decreases with the compression rate. This phenomenon is partially explained by the smoothing effect of L_2-driven compression, and is therefore less severe in Figure 5.9. Nonetheless, it remains present and could be a sign of overfitting due to the reduced amount of training data in latent space.

Table 5.1. Inference time per state for the Euler dataset, including generation and decoding.

Method	Space	Time
simulator	pixel	$\mathcal{O}(10\,\mathrm{s})$
neural solver	pixel	56 ms
neural solver	latent	13 ms
diffusion	pixel	$\mathcal{O}(1\,\mathrm{s})$
diffusion	latent	84 ms

In terms of computational cost, although they remain slower than latent-space neural solvers, latent-space diffusion models are much faster than their pixel-space counterparts and competitive with pixel-space neural solvers (see Table 5.1). With our latent diffusion models, generating and decoding a full (100 simulation steps, 7 autoregressive steps) Euler trajectory takes 3 seconds on a single A100 GPU, compared to roughly 1 CPU-hour with the original numerical simulation [40, 79].

A final advantage of diffusion models lies in their capacity to incorporate additional information during sampling via guidance methods [44, 80–83]. For example, if partial or noisy state observations are available, we can guide the emulation such that it remains

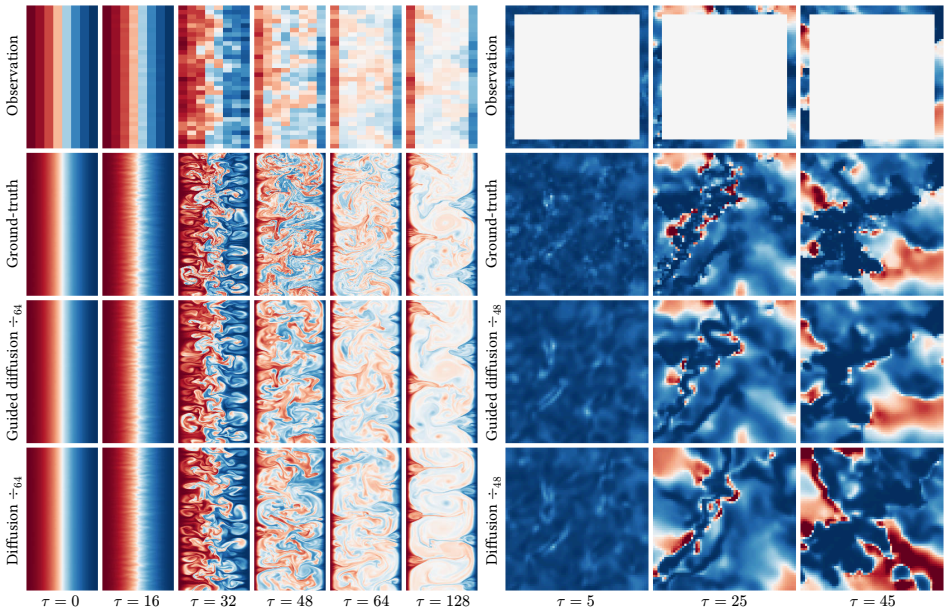


Figure 5.6. Example of guided latent-space emulation for the RB (left) and TGC (right) datasets. The observations are the states downsampled by a factor 16 for RB and a stripe along the domain boundaries for TGC. Guidance is performed using the MMPS [80] method. Thanks to the additional information in the observations, the emulation diverges less from the ground-truth.

consistent with these observations. We provide an illustrative example in Figure 5.6 where guidance is performed with the MMPS [80] method. Thanks to the additional information in the observations, the emulation diverges less from the ground-truth.

5.5 Related work

Data-driven emulation of dynamical systems has become a prominent research area [10–19] with diverse applications, including accelerating fluid simulations on uniform meshes using convolutional networks [10, 14], emulating various physics on non-uniform meshes with graph neural networks [11–13, 16], and solving partial differential equations with neural operators [15, 23, 84–86]. However, McCabe et al. [17] and Herde et al. [18] highlight the large data requirements of these methods and propose pre-training on multiple data-abundant physics before fine-tuning on data-scarce ones to improve data efficiency and generalization. Our experiments similarly suggest that large datasets are needed to train latent-space emulators.

A parallel line of work, related to reduced-order modeling [87], focuses on learning low-dimensional representations of high-dimensional system states. Within this latent space, dynamics can be emulated more efficiently [88–96]. Various embedding approaches have been explored: convolutional autoencoders for uniform meshes [90, 91], graph-based autoencoders for non-uniform meshes [92], and implicit neural representations for discretization-free states [36, 94]. Koopman operator theory [97] has also been integrated into autoencoder training to promote linear latent dynamics [93, 98]. Other approaches to enhance latent predictability include regularizing temporal derivatives [99], jointly optimizing the decoder and latent emulator [100], and self-supervised prediction [101]. While our work adopts this latent emulation paradigm, we do not impose structural biases on the latent space beside reconstruction quality.

A persistent challenge in neural emulation is ensuring temporal stability. Many models, while accurate for short-term prediction, exhibit long-term instabilities as errors accumulate, pushing the predictions out of the training data distribution [23]. Several strategies have been proposed to mitigate this issue: autoregressive unrolling during training [13, 88, 102], architectural modifications [23, 85], noise injection [14], and post-processing [20, 103]. Generative models, particularly diffusion models, have recently emerged as a promising approach to address this problem [20, 21, 24–27] as they produce statistically plausible states, even when they diverge from the ground-truth solution.

While more accurate and stable, diffusion models are computationally expensive at inference. Drawing inspiration from latent space generation in computer vision [28–34], recent studies have applied latent diffusion models to emulate dynamical systems: Gao et al. [35] address short-term precipitation forecasting, Zhou et al. [37] generate trajectories conditioned on text descriptions, Du et al. [36] generate trajectories within an implicit neural representation, and Li et al. [38] combine a state-wise autoencoder with a spatio-temporal diffusion transformer [29] for autoregressive emulation, similar to our approach. These studies report favorable or competitive results against pixel-space and deterministic baselines, consistent with our observations.

5.6 Discussion

Our results reveal key insights about latent physics emulation. First, diffusion-based emulation accuracy is surprisingly robust to latent-space compression, with performance remaining constant or even improving when autoencoder reconstruction quality significantly deteriorates. This observation is consistent with the latent generative

modeling literature [28, 58], where compression serves a dual purpose: reducing dimensionality and filtering out perceptually irrelevant patterns that might distract from semantically meaningful information. Our experiments support this hypothesis as latent-space emulators outperform their pixel-space counterparts despite using fewer parameters and requiring less training compute. Yao et al. [104] similarly demonstrate that higher compression can sometimes improve generation quality despite degrading reconstruction. While our findings seem to violate the famous data processing inequality, they are well aligned with the theory of *usable* information [106], where a learned representation can hold more $\mathcal{V}$-information from the point of view of a computationally constrained observer. Second, diffusion-based generative emulators consistently achieve higher ensemble accuracy than deterministic neural solvers while producing diverse, statistically plausible trajectories. This supports the idea that generative models mitigate distribution shift [20, 21, 24–27]. However, at the first prediction step, before distribution shift can take effect, diffusion models are already more accurate than deterministic neural solvers. This suggests an inherent modeling advantage, possibly lying in the iterative nature of diffusion sampling.

The weakness of the Kolmogorov complexity theory is that it focuses on Shannon information and ignores computability.

— Ilya Sutskever (2023)

Despite the finite number of datasets, we believe that our findings are likely to generalize well across the broader spectrum of fluid dynamics. The Euler, RB and TGC datasets represent distinct fluid regimes that cover many key challenges in dynamical systems emulation: nonlinearities, multi-scale interactions, and complex spatio-temporal patterns. In addition, previous studies [35–38] come to similar conclusions for other fluid dynamics problems. However, we exercise caution about extending these conclusions beyond fluids. Systems governed by fundamentally different physics, such as chemical or quantum phenomena, may respond unpredictably to latent compression. Probing these boundaries represents an important direction for future research. Our empirical findings also prompt the need for theoretical explanations, which we leave to future work.

Apart from datasets, if compute resources were not a limiting factor, our study could be extended along several dimensions, although we anticipate that additional experiments would not fundamentally alter our conclusions. First, we could investigate techniques for improving the structure of the latent representation, such as incorporating Koopman-inspired losses [93, 98], regularizing temporal derivatives [99], or training shallow auxiliary decoders [104, 107]. Second, we could probe the behavior of different embedding strategies under high compression, including spatio-temporal embeddings [36, 37, 108], implicit neural representations [36, 94], and masked autoencoders [107, 109]. Third, we could add the capability to trade speed for accuracy, analogous to running numerical solvers at finer resolutions, by training an autoencoder with an adaptive latent dimensionality [110–112]. Fourth, we could study the effects of autoencoder and emulator capacity by scaling either up or down their number of trainable parameters. Each of these directions represents a substantial computational investment, particularly given the scale of our datasets and models, but would help establish best practices for latent-space emulation.

Nevertheless, our findings lead to clear recommendations for practitioners wishing to implement physics emulators. First, try latent-space approaches before pixel-space emulation. The former offer reduced computational requirements, lower memory footprint, and comparable or better accuracy across a wide range of compression rates. Second, prefer diffusion-based emulators over deterministic neural solvers. Latent diffusion models provide more accurate, diverse and stable long-term trajectories, while narrowing the inference speed gap significantly.

Our experiments, however, reveal important considerations about dataset scale when training latent-space emulators. The decreasing spread-skill ratio observed at higher compression rates suggests potential overfitting. This makes intuitive sense: as

compression increases, the effective size of the dataset in latent space decreases, making overfitting more likely at fixed model capacity. Benchmarking latent emulators on smaller (10-100 GB) datasets like those used by Kohl et al. [21] could therefore yield misleading results. In addition, because the latent space is designed to preserve pixel space content, observing overfitting in this compressed representation suggests that pixel-space models encounter similar issues that remain undetected. This points towards the need for large training datasets or mixtures of datasets used to pre-train emulators before fine-tuning on targeted physics, as advocated by McCabe et al. [17] and Herde et al. [18].

Acknowledgments

We thank Géraud Krawezik and the Scientific Computing Core at the Flatiron Institute, a division of the Simons Foundation, for the compute facilities and support. We gratefully acknowledge use of the research computing resources of the Empire AI Consortium, Inc., with support from the State of New York, the Simons Foundation, and the Secunda Family Foundation. Polymathic AI acknowledges funding from the Simons Foundation and Schmidt Sciences, LLC. François Rozet is a research fellow of the F.R.S.-FNRS (Belgium) and acknowledges its financial support.

References

[1] Jürgen Schmidhuber. "Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes". In *Anticipatory Behavior in Adaptive Learning Systems*. Springer, 2009.

[2] François Rozet et al. "Lost in Latent Space: An Empirical Study of Latent Diffusion Models for Physics Emulation". In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

[3] ECMWF. "IFS documentation CY49R1 - part III: Dynamics and numerical procedures". In *IFS Documentation CY49R1*. ECMWF, 2024.

[4] Jongil Han and Hua-Lu Pan. "Revision of Convection and Vertical Diffusion Schemes in the NCEP Global Forecast System". In *Weather and Forecasting* 26.4 (2011).

[5] A. J. Hundhausen and R. A. Gentry. "Numerical simulation of flare-generated disturbances in the solar wind". In *Journal of Geophysical Research* 74.11 (1969).

[6] John T. Mariska, A. Gordon Emslie, and Peng Li. "Numerical Simulations of Impulsively Heated Solar Flares". In *The Astrophysical Journal* 341 (1989).

[7] Chi Wang et al. "Magnetohydrodynamics (MHD) numerical simulations on the interaction of the solar wind with the magnetosphere: A review". In *Science China Earth Sciences* 56.7 (2013).

[8] Yuri N. Dnestrovskii and Dimitri P. Kostomarov. "Numerical Simulation of Plasmas". Springer, 1986.

[9] Yildirim Suzen et al. "Numerical Simulations of Plasma Based Flow Control Applications". In *35th AIAA Fluid Dynamics Conference and Exhibit*. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2005.

[10] Jonathan Tompson et al. "Accelerating Eulerian Fluid Simulation With Convolutional Networks". In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017.

[11] Alvaro Sanchez-Gonzalez et al. "Learning to Simulate Complex Physics with Graph Networks". In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[12] Tobias Pfaff et al. "Learning Mesh-Based Simulation with Graph Networks". In *International Conference on Learning Representations*. 2021.

[13] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. "Message Passing Neural PDE Solvers". In *International Conference on Learning Representations*. 2022.

[14] Kim Stachenfeld et al. "Learned Simulators for Turbulence". In *International Conference on Learning Representations*. 2022.

[15] Nikola Kovachki et al. "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs". In *Journal of Machine Learning Research* 24.89 (2023).

[16] Remi Lam et al. "Learning skillful medium-range global weather forecasting". In *Science* 382.6677 (2023).

[17] Michael McCabe et al. "Multiple Physics Pretraining for Spatiotemporal Surrogate Models". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[18] Maximilian Herde et al. “Poseidon: Efficient Foundation Models for PDEs”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[19] Rudy Morel, Jiequn Han, and Edouard Oyallon. “DISCO: learning to DISCover an evolution Operator for multi-physics-agnostic prediction”. 2025.

[20] Phillip Lippe et al. “PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers”. In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[21] Georg Kohl, Liwei Chen, and Nils Thuerey. “Benchmarking Autoregressive Conditional Diffusion Models for Turbulent Flow Simulation”. In *AI for Science Workshop (ICML)*. 2024.

[22] Björn List, Li-Wei Chen, and Nils Thuerey. “Learned turbulence modelling with differentiable fluid solvers: physics-based loss functions and optimisation horizons”. In *Journal of Fluid Mechanics* 949 (2022).

[23] Michael McCabe et al. “Towards Stability of Autoregressive Neural Operators”. In *Transactions on Machine Learning Research* (2023).

[24] Salva Cachay et al. “DYffusion: A Dynamics-informed Diffusion Model for Spatiotemporal Forecasting”. In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[25] Aliaksandra Shysheya et al. “On conditional diffusion models for PDE simulations”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[26] Jiahe Huang et al. “DiffusionPDE: Generative PDE-Solving under Partial Observation”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[27] Ilan Price et al. “Probabilistic weather forecasting with machine learning”. In *Nature* 637.8044 (2025).

[28] Robin Rombach et al. “High-Resolution Image Synthesis With Latent Diffusion Models”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[29] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In *IEEE/CVF International Conference on Computer Vision*. 2023.

[30] Patrick Esser et al. “Scaling Rectified Flow Transformers for High-Resolution Image Synthesis”. 2024.

[31] Tero Karras et al. “Analyzing and Improving the Training Dynamics of Diffusion Models”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

[32] Enze Xie et al. “SANA: Efficient High-Resolution Text-to-Image Synthesis with Linear Diffusion Transformers”. In *International Conference on Learning Representations*. 2025.

[33] Junyu Chen et al. “Deep Compression Autoencoder for Efficient High-Resolution Diffusion Models”. In *International Conference on Learning Representations*. 2025.

[34] Adam Polyak et al. “Movie Gen: A Cast of Media Foundation Models”. 2024.

[35] Zhihan Gao et al. “PreDiff: Precipitation Nowcasting with Latent Diffusion Models”. In *Advances in Neural Information Processing Systems*. Vol. 37. 2023.

[36] Pan Du et al. “Conditional neural field latent diffusion model for generating spatiotemporal turbulence”. In *Nature Communications* 15.1 (2024).

[37] Anthony Zhou et al. “Text2PDE: Latent Diffusion Models for Accessible Physics Simulation”. In *International Conference on Learning Representations*. 2025.

[38] Zijie Li, Anthony Zhou, and Amir Barati Farimani. "Generative Latent Neural PDE Solver using Flow Matching". 2025.

[39] Gérôme Andry et al. "Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2025.

[40] Ruben Ohana et al. "The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[41] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[42] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[43] Yang Song and Stefano Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[44] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In *International Conference on Learning Representations*. 2021.

[45] Brian D. O. Anderson. "Reverse-time diffusion equation models". In *Stochastic Processes and their Applications* 12.3 (1982).

[46] Simo Särkkä and Arno Solin. "Applied Stochastic Differential Equations". Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[47] Shanchuan Lin et al. "Common Diffusion Noise Schedules and Sample Steps are Flawed". In *IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024.

[48] Xingchao Liu, Chengyue Gong, and Qiang Liu. "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In *International Conference on Learning Representations*. 2023.

[49] Yaron Lipman et al. "Flow Matching for Generative Modeling". In 2023.

[50] Aapo Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". In *Journal of Machine Learning Research* (2005).

[51] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". In *Neural Computation* (2011).

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In *International Conference on Learning Representations*. 2021.

[53] Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models". In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[54] M. C. K. Tweedie. "Functions of a statistical variate with given means, with special reference to Laplacian distributions". In *Mathematical Proceedings of the Cambridge Philosophical Society* (1947).

[55] Bradley Efron. "Tweedie's Formula and Selection Bias". In *Journal of the American Statistical Association* (2011).

[56] Kwanyoung Kim and Jong Chul Ye. "Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[57] Chenlin Meng et al. “Estimating High Order Gradients of the Data Distribution by Denoising”. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[58] Sander Dieleman. “Generative modelling in latent space”. 2025.

[59] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

[60] Ian J. Goodfellow et al. “Generative Adversarial Networks”. 2014.

[61] Patrick Esser, Robin Rombach, and Bjorn Ommer. “Taming Transformers for High-Resolution Image Synthesis”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

[62] Xi-Lin Li. “Preconditioned Stochastic Gradient Descent”. In *IEEE Transactions on Neural Networks and Learning Systems* 29.5 (2018).

[63] Vineet Gupta, Tomer Koren, and Yoram Singer. “Shampoo: Preconditioned Stochastic Tensor Optimization”. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018.

[64] Nikhil Vyas et al. “SOAP: Improving and Stabilizing Shampoo using Adam for Language Modeling”. In *International Conference on Learning Representations*. 2025.

[65] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In *International Conference on Learning Representations*. 2015.

[66] Lucas Nestler and François Rozet. “HeavyBall: Efficient optimizers”. 2022.

[67] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. “MCVD - Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation”. In *Advances in Neural Information Processing Systems*. Vol. 35. 2022.

[68] Alex Henry et al. “Query-Key Normalization for Transformers”. In *Findings of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[69] Jianlin Su et al. “RoFormer: Enhanced transformer with Rotary Position Embedding”. In *Neurocomputing* 568 (2024).

[70] Byeongho Heo et al. “Rotary Position Embedding for Vision Transformer”. In *European Conference on Computer Vision*. Springer Nature Switzerland, 2025.

[71] Zhanchao Zhou et al. “Value Residual Learning”. 2024.

[72] Zilong Huang et al. “CCNet: Criss-Cross Attention for Semantic Segmentation”. In *IEEE/CVF International Conference on Computer Vision*. 2019.

[73] Jonathan Ho et al. “Axial Attention in Multidimensional Transformers”. 2019.

[74] Ali Hassani et al. “Neighborhood Attention Transformer”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[75] Qinsheng Zhang and Yongxin Chen. “Fast Sampling of Diffusion Models with Exponential Integrator”. In *International Conference on Learning Representations*. 2023.

[76] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In *International Conference on Learning Representations*. 2021.

[77] V. Fortin et al. “Why Should Ensemble Spread Match the RMSE of the Ensemble Mean?” In *Journal of Hydrometeorology* 15.4 (2014).

[78] Hugh L. Dryden. "A Review of the Statistical Theory of Turbulence". In *Quarterly of Applied Mathematics* 1.1 (1943).

[79] Kyle T. Mandli et al. "Clawpack: building an open source ecosystem for solving hyperbolic PDEs". In *PeerJ Computer Science* 2 (2016).

[80] François Rozet et al. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[81] Jonathan Ho et al. "Video Diffusion Models". In *ICLR Workshop on Deep Generative Models for Highly Structured Data*. 2022.

[82] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In *International Conference on Learning Representations*. 2023.

[83] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[84] Zongyi Li et al. "Fourier Neural Operator for Parametric Partial Differential Equations". In *International Conference on Learning Representations*. 2021.

[85] Bogdan Raonic et al. "Convolutional Neural Operators for robust and accurate learning of PDEs". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[86] Zhongkai Hao et al. "GNOT: A General Neural Operator Transformer for Operator Learning". In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[87] Peter Benner, Serkan Gugercin, and Karen Willcox. "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems". In *SIAM Review* 57.4 (2015).

[88] Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. "Deep learning for universal linear embeddings of nonlinear dynamics". In *Nature Communications* 9.1 (2018).

[89] Hugo F. S. Lui and William R. Wolf. "Construction of reduced-order models for fluid flows using deep feedforward neural networks". In *Journal of Fluid Mechanics* 872 (2019).

[90] S. Wiewel, M. Becher, and N. Thuerey. "Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow". In *Computer Graphics Forum* 38.2 (2019).

[91] Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. "Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders". In *Physics of Fluids* 33.3 (2021).

[92] Xu Han et al. "Predicting Physics in Mesh-reduced Space with Temporal Attention". In *International Conference on Learning Representations*. 2022.

[93] Nicholas Geneva and Nicholas Zabaras. "Transformers for modeling physical systems". In *Neural Networks* 146 (2022).

[94] Peter Yichen Chen et al. "CROM: Continuous Reduced-Order Modeling of PDEs Using Implicit Neural Representations". In *International Conference on Learning Representations*. 2023.

[95] AmirPouya Hemmasian and Amir Barati Farimani. "Reduced-order modeling of fluid flows with transformers". In *Physics of Fluids* 35.5 (2023).

[96] Zijie Li et al. "Latent neural PDE solver: A reduced-order modeling framework for partial differential equations". In *Journal of Computational Physics* 524 (2025).

[97] B. O. Koopman. “Hamiltonian Systems and Transformation in Hilbert Space”. In *Proceedings of the National Academy of Sciences* 17.5 (1931).

[98] Enoch Yeung, Soumya Kundu, and Nathan Hodas. “Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems”. In *American Control Conference*. 2019.

[99] Xiaoyu Xie, Saviz Mowlavi, and Mouhacine Benosman. “Smooth and Sparse Latent Dynamics in Operator Learning with Jerk Regularization”. 2024.

[100] Francesco Regazzoni et al. “Learning the intrinsic dynamics of spatio-temporal processes through Latent Dynamics Networks”. In *Nature Communications* 15.1 (2024).

[101] Adrien Bardes et al. “Revisiting Feature Prediction for Learning Visual Representations from Video”. In *Transactions on Machine Learning Research* (2024).

[102] Nicholas Geneva and Nicholas Zabaras. “Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks”. In *Journal of Computational Physics* 403 (2020).

[103] Daniel E. Worrall et al. “Spectral Shaping for Neural PDE Surrogates”. 2024.

[104] Jingfeng Yao, Bin Yang, and Xinggang Wang. “Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models”. 2025.

[105] Ilya Sutskever. “An Observation on Generalization”. 2023.

[106] Yilun Xu et al. “A Theory of Usable Information under Computational Constraints”. In *International Conference on Learning Representations*. 2020.

[107] Hao Chen et al. “Masked Autoencoders Are Effective Tokenizers for Diffusion Models”. 2025.

[108] Lijun Yu et al. “MAGVIT: Masked Generative Video Transformer”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[109] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[110] Alekh Karkada Ashok and Nagaraju Palani. “Autoencoders with Variable Sized Latent Vector for Image Compression”. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.

[111] Chi-Hieu Pham, Saïd Ladjal, and Alasdair Newson. “PCA-AE: Principal Component Analysis Autoencoder for Organising the Latent Space of Generative Networks”. In *Journal of Mathematical Imaging and Vision* 64.5 (2022).

[112] Roman Bachmann et al. “FlexTok: Resampling Images into 1D Token Sequences of Flexible Length”. In *Proceedings of the 42nd International Conference on Machine Learning*. 2025.

[113] Keaton J. Burns et al. “Dedalus: A flexible framework for numerical simulations with spectral methods”. In *Physical Review Research* 2.2 (2020).

[114] Keiya Hirashima et al. “3D-Spatiotemporal forecasting the expansion of supernova shells using deep learning towards high-resolution galaxy simulations”. In *Monthly Notices of the Royal Astronomical Society* 526.3 (2023).

[115] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[116] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”. In *Neural Networks*. Special issue on deep reinforcement learning 107 (2018).

[117] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. “Layer Normalization”. 2016.

[118] Nicolas Bonneel et al. “Sliced and Radon Wasserstein Barycenters of Measures”. In *Journal of Mathematical Imaging and Vision* (2015).

[119] Soheil Kolouri et al. “Generalized Sliced Wasserstein Distances”. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[120] Tung Nguyen et al. “PhysiX: A Foundation Model for Physics Simulations”. 2025.

[121] Zhikai Wu et al. “TANTE: Time-Adaptive Operator Learning via Neural Taylor Expansion”. 2025.

[122] Payel Mukhopadhyay et al. “Controllable Patching for Compute-Adaptive Surrogate Modeling of Partial Differential Equations”. 2025.

[123] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In *Journal of Machine Learning Research* 9.86 (2008).

[124] Gabriel Peyré and Marco Cuturi. “Computational Optimal Transport: With Applications to Data Science”. In *Foundations and Trends in Machine Learning* 11.5-6 (2019).

5.A Spread / Skill

The skill [27, 77] of an ensemble of K particles v_k is defined as the RMSE of the ensemble mean

$$\text{Skill} = \sqrt{\left\langle \left(u - \frac{1}{K} \sum_{k=1}^{K} v_k \right)^2 \right\rangle} \tag{5.11}$$

where $\langle \cdot \rangle$ denotes the spatial mean operator. The spread is defined as the ensemble standard deviation

$$\text{Spread} = \sqrt{\left\langle \frac{1}{K-1} \sum_{j=1}^{K} \left(v_j - \frac{1}{K} \sum_{k=1}^{K} v_k \right)^2 \right\rangle} . \tag{5.12}$$

Under these definitions and the assumption of a perfect forecast where ensemble particles are exchangeable, Fortin et al. [77] show that

$$\text{Skill} \approx \sqrt{\frac{K+1}{K}} \, \text{Spread} \ . \tag{5.13}$$

This motivates the use of the (corrected) spread-skill ratio as a metric. Intuitively, if the ratio is smaller than one, the ensemble is biased or under-dispersed. If the ratio is larger than one, the ensemble is over-dispersed. It should be noted however, that a spread-skill ratio of 1 is a necessary but not sufficient condition for a perfect forecast.

5.B Experiment details

Datasets For all datasets, each field is standardized with respect to its mean and variance over the training set. For Euler, the non-negative scalar fields (energy, density, pressure) are transformed with $x \mapsto \log(x+1)$ before standardization. For TGC, the non-negative scalar fields (density, pressure, temperature) are transformed with $x \mapsto \log(x + 10^{-6})$ before standardization. When the states are illustrated graphically, as in Figure 5.1, we represent the density field for Euler, the buoyancy field for RB, and a slice of the temperature field for TGC.

Table 5.2. Details of the selected datasets. We refer the reader to Ohana et al. [40] for more information.

	Euler Multi-Quadrants	Rayleigh-Bénard	Turbulence Gravity Cooling
Software	Clawpack [79]	Dedalus [113]	ASURA-FDPS [114]
Size	5243 GB	367 GB	849 GB
Fields	energy, density, pressure, velocity	buoyancy, pressure, momentum	density, pressure, temperature, velocity
Channels C_{pixel}	5	4	6
Resolution	512×512	512×128	$64 \times 64 \times 64$
Discretization	Uniform	Chebyshev	Uniform
Trajectories	10000	1750	2700
Time steps L	100	200	50
Stride Δ	4	1	1
θ	heat capacity γ, boundary conditions	Rayleigh number, Prandtl number	hydrogen density ρ_0, temperature T_0, metallicity Z

Autoencoders The encoder E_ψ and decoder D_ψ are convolutional networks with residual blocks [115], SiLU [116] activation functions and layer normalization [117]. The output of the encoder is transformed with a saturating function (see Section 5.3). We provide a schematic illustration of the autoencoder architecture in Figure 5.7. Following McCabe et al. [17], we use a field-weighted loss ℓ, and choose the variance-normalized MSE (VMSE)

$$\text{VMSE}(u, v) = \frac{\left\langle (u-v)^2 \right\rangle}{\left\langle (u - \langle u \rangle)^2 \right\rangle + \epsilon} \tag{5.14}$$

averaged over fields, where $\epsilon = 10^{-2}$ mitigates training instabilities. We train the encoder and decoder jointly for 1024×256 steps of the PSGD [62] optimizer. To mitigate overfitting we use random spatial axes permutations, flips and rolls as data augmentation. Each autoencoder takes 1 (RB), 2 (Euler) or 4 (TGC) days to train on 8 H100 GPUs. Other hyperparameters are provided in Table 5.3.

Caching The entire dataset is encoded with each trained autoencoder and the resulting latent trajectories are cached permanently on disk. The latter can then be used to train latent-space emulators, without needing to load and encode high-dimensional samples on the fly. Depending on hardware and data dimensionality, this approach can make a huge difference in I/O efficiency.

Table 5.3. Hyperparameters for the autoencoders.

	Euler & RB	TGC
Architecture	Conv	Conv
Parameters	3.1×10^8	7.2×10^8
Pixel shape	$C_{\text{pixel}} \times H \times W$	$C_{\text{pixel}} \times H \times W \times Z$
Latent shape	$C_{\text{latent}} \times \frac{H}{32} \times \frac{W}{32}$	$C_{\text{latent}} \times \frac{H}{8} \times \frac{W}{8} \times \frac{Z}{8}$
Residual blocks per level	(3, 3, 3, 3, 3, 3)	(3, 3, 3, 3)
Channels per level	(64, 128, 256, 512, 768, 1024)	(64, 256, 512, 1024)
Kernel size	3×3	$3 \times 3 \times 3$
Activation	SiLU	SiLU
Normalization	LayerNorm	LayerNorm
Dropout	0.05	0.05
Loss	VMSE	VMSE
Optimizer	PSGD	PSGD
Learning rate	10^{-5}	10^{-5}
Weight decay	0.0	0.0
Scheduler	cosine	cosine
Gradient norm clipping	1.0	1.0
Batch size	64	64
Steps per epoch	256	256
Epochs	1024	1024
GPUs	8	8

Table 5.4. Short ablation study on the autoencoder architecture and training configurations. We pick the Rayleigh-Bénard dataset and an architecture with 64 latent channels to perform this study. The two major modifications that we propose are (1) the initialization of the downsampling and upsampling layers near identity, inspired by Chen et al. [33], and (2) the use of a preconditioned optimizer, PSGD [62], instead of Adam [65]. We report the mean absolute error (MAE) on the validation set during training. The combination of both proposed modifications leads to order(s) of magnitude faster convergence.

Optimizer	Id. init	Epoch			Time
		10	100	1000	
Adam	w/o	0.065	0.029	0.017	19 h
Adam	w/	0.039	0.023	0.014	19 h
PSGD	w/	0.023	0.015	0.011	25 h

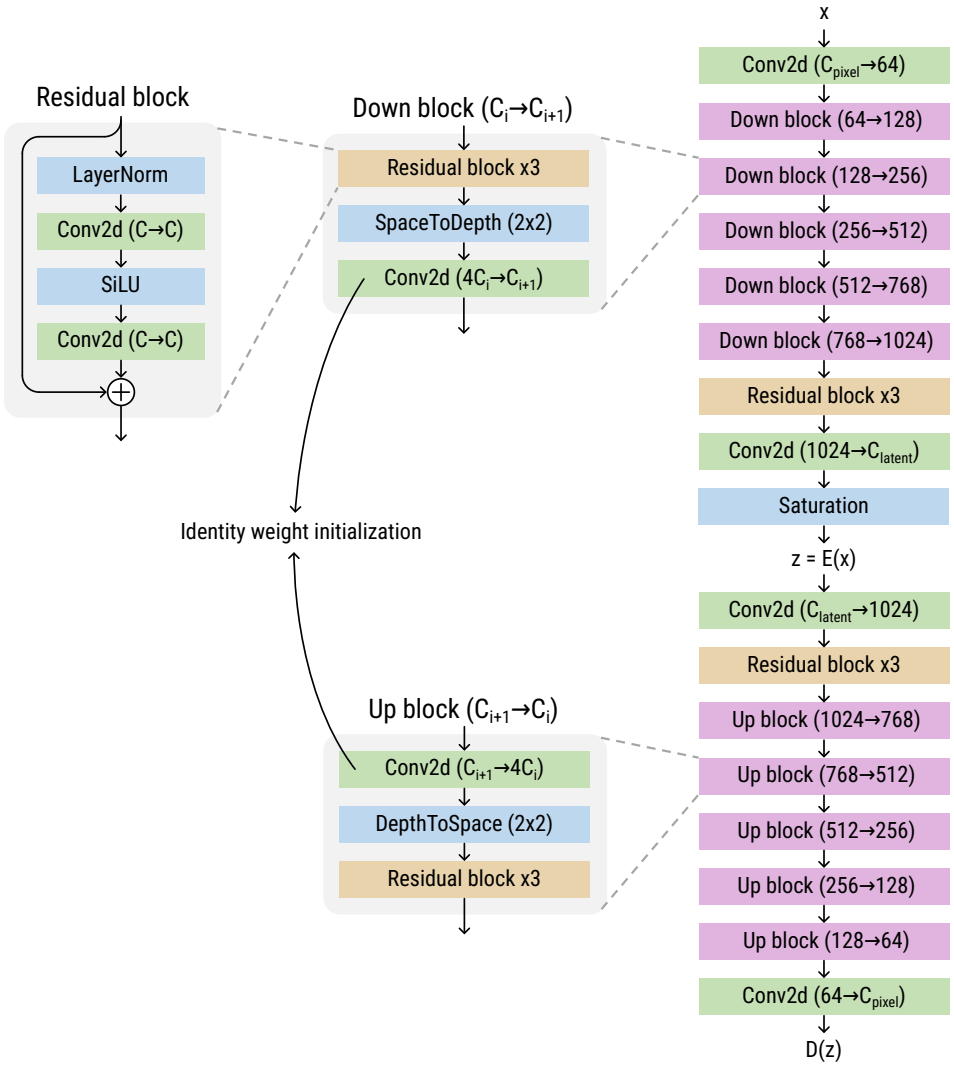


Figure 5.7. Schematic representation of the autoencoder architecture. Downsampling (resp. upsampling) is performed with a space-to-depth (resp. depth-to-space) operation followed (resp. preceded) with a convolution initialized near identity.

Emulators The denoiser d_ϕ and neural solver f_ϕ are transformers with query-key normalization [68], rotary positional embedding (RoPE) [69, 70], and value residual learning [71]. The 16 blocks are modulated by the simulation parameters θ and the diffusion time t, as in diffusion transformers [29]. We train the emulator for 4096×64 steps of the Adam [65] optimizer. Each latent-space emulator takes 2 (RB) or 5 (Euler, TGC) days to train on 8 H100 GPUs. Each pixel-space emulator takes 5 (RB) or 10 (Euler) days to train on 16 H100 GPUs. We do not train a pixel-space emulator for TGC. Other hyperparameters are provided in Table 5.5

During training we randomly sample the binary mask b. The number of context elements c follows a Poisson distribution $\text{Pois}(\lambda = 2)$ truncated between 1 and n. Hence, the masks b take the form

$$b = (\underbrace{1, \dots, 1}_{c}, 0, \dots, 0) \tag{5.15}$$

implicitly defining a distribution $p(b)$.

Table 5.5. Hyperparameters for the emulators.

	Latent-space	Pixel-space
Architecture	Transformer	Transformer
Parameters	2.2×10^8	8.6×10^8
Input shape	$C_{\text{latent}} \times (n+1) \times \frac{H}{32} \times \frac{W}{32}$	$C_{\text{pixel}} \times (n+1) \times H \times W$
Patch size	$1 \times 1 \times 1$	$1 \times 16 \times 16$
Tokens	$(n+1) \times \frac{H}{32} \times \frac{W}{32}$	$(n+1) \times \frac{H}{16} \times \frac{W}{16}$
Embedding size	1024	2048
Blocks	16	16
Positional embedding	Absolute + RoPE	Absolute + RoPE
Activation	SiLU	SiLU
Normalization	LayerNorm	LayerNorm
Dropout	0.05	0.05
Optimizer	Adam	Adam
Learning rate	10^{-4}	10^{-4}
Weight decay	0.0	0.0
Scheduler	cosine	cosine
Gradient norm clipping	1.0	1.0
Batch size	256	256
Steps per epoch	64	64
Epochs	4096	4096
GPUs	8	16

Evaluation For each dataset, we randomly select 64 trajectories $x^{0:L}$ with various parameters θ in the test set. For each latent-space emulator, we encode the initial state $z^0 = E_\psi(x^0)$ and produce 16 distinct autoregressive rollouts $z^{1:L}$. For the diffusion models, sampling is performed with 16 steps of the 3rd order Adams-Bashforth multi-step integration method [75]. The metrics (VRMSE, power spectrum RMSE, spread-skill ratio) are then measured between the predicted states $\hat{x}^i = D_\psi(z^i)$ and the ground-truth states x^i or the autoencoded states $D_\psi(E_\psi(x^i))$.

5.C Additional emulation results

Table 5.6. Average VRMSE of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the Euler, RB and TGC datasets. Increasing the compression rate has a clear impact on reconstruction quality, but does not degrade significantly (Euler, TGC) and sometimes improves (RB) the accuracy of diffusion models.

Method	Euler			
	÷	1:20	21:60	61:100
autoencoder	80	0.011	0.014	0.020
	320	0.023	0.041	0.061
	1280	0.060	0.107	0.144
diffusion	80	0.075	0.199	0.395
	320	0.070	0.192	0.371
	1280	0.093	0.217	0.400
neural solver	1	0.138	0.397	1.102
	80	0.077	0.232	0.500
	320	0.080	0.232	0.476
	1280	0.137	0.314	0.592

Method	RB			
	÷	1:20	21:60	61:180
autoencoder	64	0.023	0.033	0.019
	256	0.039	0.064	0.042
	1024	0.070	0.124	0.092
diffusion	64	0.171	0.582	0.704
	256	0.141	0.509	0.683
	1024	0.146	0.457	0.670
neural solver	1	0.185	0.681	0.918
	64	0.244	0.761	0.968
	256	0.197	0.716	0.945
	1024	0.195	0.665	0.903

Method	TGC			
	÷	1:10	11:20	21:50
autoencoder	48	0.151	0.116	0.129
	192	0.229	0.175	0.189
	768	0.338	0.272	0.276
diffusion	48	0.296	0.522	0.673
	192	0.342	0.527	0.665
	768	0.425	0.575	0.694
neural solver	48	0.302	0.599	0.826
	192	0.361	0.632	0.835
	768	0.462	0.710	0.920

Table 5.7. Average VRMSE of latent-space emulation at different context lengths (c) and lead time horizons for the Euler, RB and TGC datasets. We can test different context lengths without retraining as our models were trained for different conditioning tasks (see Section 5.3). Perhaps surprisingly, context lengths does not have a significant impact on emulation accuracy.

Method	Euler			
	c	1:20	21:60	61:100
diffusion	1	0.085	0.204	0.393
	2	0.074	0.200	0.383
	3	0.078	0.203	0.389
neural solver	1	0.108	0.266	0.526
	2	0.092	0.253	0.513
	3	0.094	0.260	0.529

Method	RB			
	c	1:20	21:60	61:180
diffusion	1	0.152	0.510	0.683
	2	0.150	0.511	0.685
	3	0.157	0.527	0.689
neural solver	1	0.208	0.705	0.932
	2	0.209	0.708	0.943
	3	0.220	0.728	0.940

Method	TGC			
	c	1:10	11:20	21:50
diffusion	1	0.362	0.550	0.681
	2	0.351	0.535	0.669
	3	0.350	0.539	0.683
neural solver	1	0.376	0.632	0.837
	2	0.371	0.641	0.855
	3	0.378	0.669	0.888

Table 5.8. Average power spectrum RMSE of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the Euler dataset. The high-frequency content of diffusion-based emulators is limited by the autoencoder's reconstruction capabilities.

Method	÷	Low			Mid			High		
		1:20	21:60	61:100	1:20	21:60	61:100	1:20	21:60	61:100
autoencoder	80	0.001	0.001	0.001	0.006	0.008	0.014	0.072	0.069	0.096
	320	0.002	0.003	0.004	0.022	0.047	0.085	0.112	0.141	0.240
	1280	0.009	0.017	0.025	0.074	0.167	0.264	0.240	0.355	0.577
diffusion	80	0.017	0.063	0.168	0.054	0.100	0.178	0.112	0.116	0.184
	320	0.014	0.058	0.157	0.052	0.102	0.171	0.128	0.155	0.275
	1280	0.019	0.065	0.163	0.096	0.187	0.300	0.246	0.349	0.569
neural solver	1	0.046	0.128	0.339	0.227	0.297	0.754	0.821	0.984	2.666
	80	0.021	0.074	0.212	0.085	0.151	0.245	0.164	0.173	0.249
	320	0.020	0.075	0.204	0.074	0.144	0.234	0.151	0.169	0.271
	1280	0.045	0.116	0.274	0.131	0.227	0.349	0.283	0.345	0.545

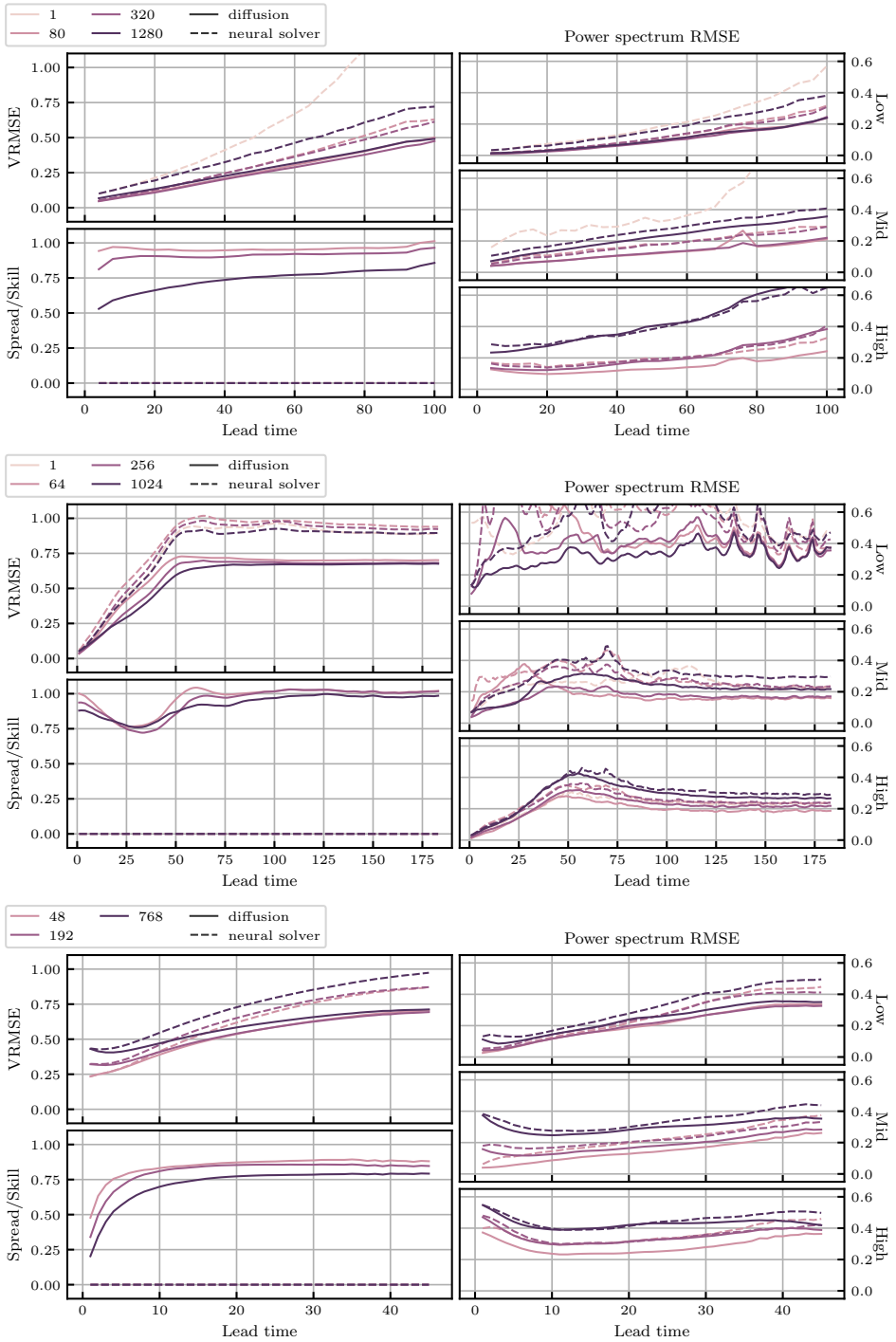


Figure 5.8. Average evaluation metrics of latent-space emulation for the Euler (top), RB (center) and TGC (bottom) datasets. As expected from imperfect emulators, the emulation error grows with the lead time. However, increasing the compression rate does not degrade (Euler, TGC) and sometimes improves (RB) the accuracy of diffusion models. The spread-skill ratio [27, 77] drops slightly with the compression rate, which could be a sign of overfitting. Diffusion-based emulators are consistently more accurate than neural solvers.

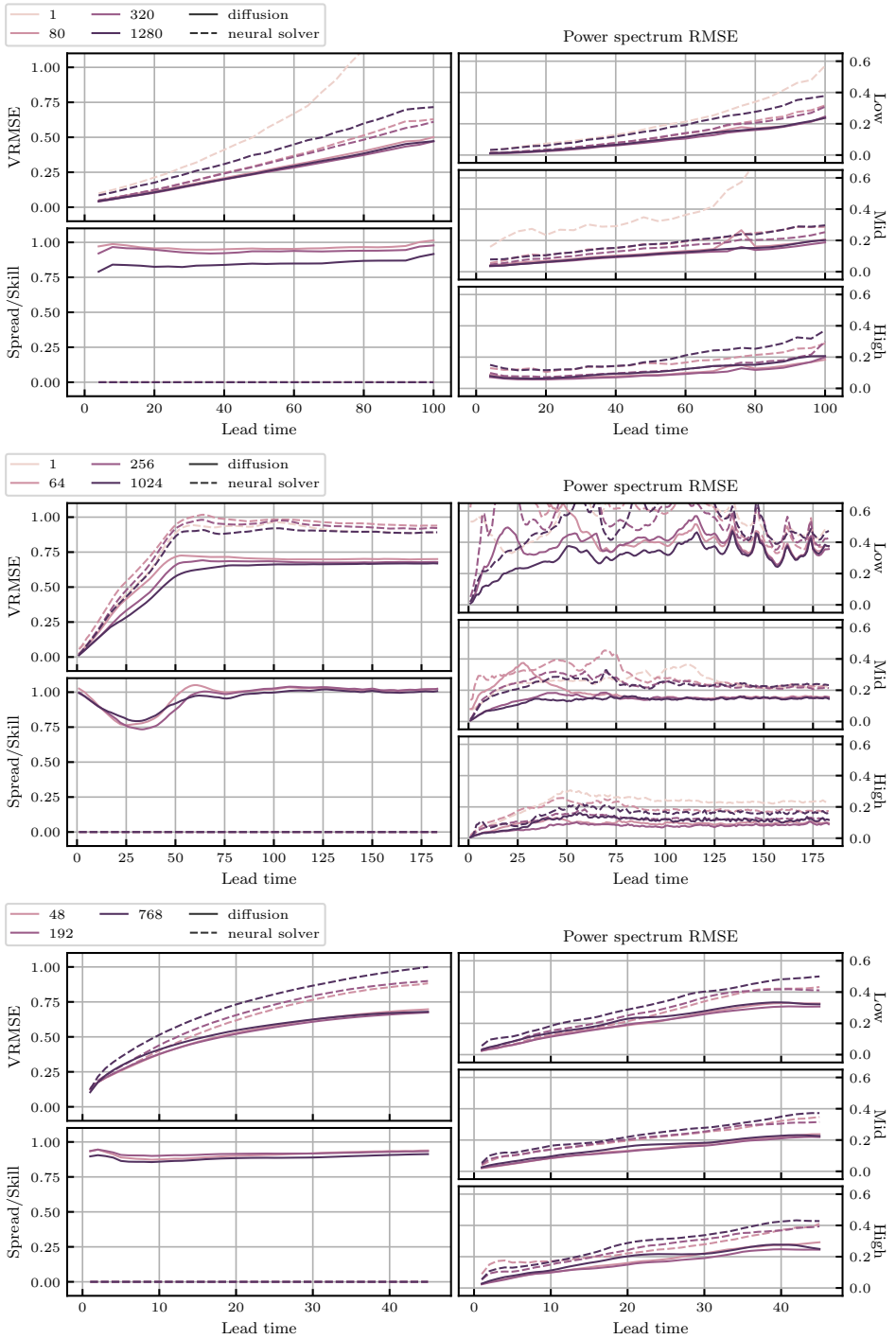


Figure 5.9. Average evaluation metrics of latent-space emulation relative to the autoencoded states $D_\psi(E_\psi(x^i))$ for the Euler (top), RB (center) and TGC (bottom) datasets. As expected from imperfect emulators, the emulation error grows with the lead time. However, increasing the compression rate does not degrade (Euler, TGC) and sometimes improves (RB) the accuracy of diffusion models. The spread-skill ratio [27, 77] drops slightly with the compression rate, which could be a sign of overfitting. Diffusion-based emulators are consistently more accurate than neural solvers.

Table 5.9. Average power spectrum RMSE of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the Rayleigh-Bénard dataset. The high-frequency content of diffusion-based emulators is limited by the autoencoder's reconstruction capabilities.

Method	÷	Low			Mid			High		
		1:20	21:60	61:180	1:20	21:60	61:180	1:20	21:60	61:180
autoencoder	64	0.043	0.004	0.001	0.011	0.013	0.012	0.026	0.159	0.148
	256	0.061	0.011	0.004	0.028	0.080	0.075	0.050	0.220	0.212
	1024	0.121	0.033	0.018	0.063	0.186	0.197	0.076	0.294	0.294
diffusion	64	1.751	0.850	0.386	0.197	0.266	0.159	0.054	0.220	0.199
	256	0.328	0.399	0.396	0.084	0.195	0.177	0.065	0.239	0.232
	1024	0.193	0.292	0.344	0.095	0.243	0.240	0.083	0.314	0.297
neural solver	1	0.467	0.520	0.650	0.151	0.255	0.264	0.076	0.232	0.242
	64	3.625	0.915	0.566	0.268	0.351	0.275	0.099	0.279	0.257
	256	0.575	0.675	0.526	0.165	0.317	0.264	0.091	0.275	0.257
	1024	0.285	0.496	0.560	0.152	0.338	0.321	0.090	0.320	0.326

Table 5.10. Average power spectrum RMSE of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the TGC dataset. The high-frequency content of diffusion-based emulators is limited by the autoencoder's reconstruction capabilities.

Method	÷	Low			Mid			High		
		1:10	11:30	31:50	1:10	11:30	31:50	1:10	11:30	31:50
autoencoder	48	0.011	0.016	0.025	0.023	0.026	0.044	0.275	0.188	0.195
	192	0.028	0.033	0.045	0.108	0.091	0.114	0.359	0.273	0.282
	768	0.072	0.068	0.080	0.285	0.235	0.254	0.454	0.476	0.367
diffusion	48	0.064	0.185	0.319	0.058	0.128	0.220	0.296	0.247	0.331
	192	0.069	0.191	0.311	0.128	0.164	0.252	0.369	0.316	0.384
	768	0.107	0.294	0.425	0.289	0.305	0.360	0.456	0.419	0.444
neural solver	48	0.070	0.221	0.424	0.110	0.197	0.324	0.357	0.320	0.427
	192	0.086	0.228	0.402	0.172	0.201	0.295	0.391	0.317	0.395
	768	0.138	0.277	0.465	0.322	0.305	0.407	0.471	0.418	0.493

Table 5.11. Average sliced earth mover's distance (SEMD) [118, 119] of the density field of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the Euler dataset. The SEMD is small and is not significantly impacted by the compression rate, especially for diffusion models. For reference, the density fields of two consecutive states x^i and x^{i+1} have a typical SEMD of 0.0025. *Why this metric?* The Euler equations are sometimes used in aerodynamics to model flow around objects and one is typically interested in the global fluid displacement. The rationale for using this metric is that a small drift in the density field would not significantly affect the (S)EMD, while it could affect point-wise metrics heavily.

Method	EMD (density field)			
	÷	1:20	21:60	61:100
autoencoder	80	0.0000	0.0000	0.0000
	320	0.0001	0.0001	0.0001
	1280	0.0002	0.0003	0.0005
diffusion	80	0.0004	0.0010	0.0023
	320	0.0003	0.0009	0.0022
	1280	0.0004	0.0010	0.0023
neural solver	1	0.0011	0.0031	0.0066
	80	0.0005	0.0012	0.0028
	320	0.0004	0.0012	0.0027
	1280	0.0008	0.0020	0.0041

Table 5.12. Average Wasserstein distance of the distribution of vertical velocity values of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the RB dataset. The Wasserstein distance is smaller for diffusion models and decreases with the compression rate. For reference, the distributions of vertical velocity values of two consecutive states x^i and x^{i+1} have a typical Wasserstein distance of 0.004. *Why this metric?* One interesting quantity in buoyancy-driven convection is the growth speed of plumes in the fluid. The distribution of the (vertical) velocity values is a good summary statistic for tracking the growth of plumes.

Method	Wasserstein (vertical velocity field)			
	÷	1:20	21:60	61:180
autoencoder	64	0.0000	0.0002	0.0002
	256	0.0001	0.0007	0.0005
	1024	0.0002	0.0020	0.0018
diffusion	64	0.0003	0.0104	0.0141
	256	0.0003	0.0092	0.0141
	1024	0.0004	0.0063	0.0139
neural solver	1	0.0003	0.0153	0.0247
	64	0.0009	0.0272	0.0223
	256	0.0007	0.0197	0.0187
	1024	0.0007	0.0157	0.0206

Table 5.13. Average Wasserstein distance of the distribution of density values of autoencoder reconstruction and latent-space emulation at different compression rates (÷) and lead time horizons for the TGC dataset. The Wasserstein distance is smaller for diffusion models, but grows significantly with the lead time, even for the autoencoder reconstruction. For reference, the distributions of density values of two consecutive states x^i and x^{i+1} have a typical Wasserstein distance of 0.01. *Why this metric?* In the interstellar medium, gravity forms clusters of matter that eventually lead to the birth of stars. The kind of clusters (compact, diffuse, or anything in between) and their proportions is of interest for domain-scientists. The distribution of the density values is a good summary statistic for clustering dynamics.

Method	Wasserstein (density field)			
	÷	1:10	11:30	31:50
autoencoder	48	0.0034	0.0048	0.0089
	192	0.0082	0.0110	0.0183
	768	0.0181	0.0236	0.0338
diffusion	48	0.0044	0.0138	0.0266
	192	0.0089	0.0172	0.0310
	768	0.0186	0.0274	0.0425
neural solver	48	0.0074	0.0253	0.0524
	192	0.0091	0.0171	0.0329
	768	0.0220	0.0296	0.0492

Table 5.14. Average VRMSE results from various studies using TheWell [40] datasets. Even though our latent neural solvers (LNSs) and latent diffusion models (LDMs) outperform most published baselines, we emphasize that we do not position our models as state-of-the-art, due to the discrepancies in parameter count, training and evaluation. Notably, the U-Net and FNO baselines trained by Ohana et al. [40] are much smaller than other models, and their hyper-parameters were not tuned.

Source	Method	Dataset	Parameters	Lead time	VRMSE
Ohana et al. [40]	FNO	Euler	2×10^7	6:12	1.13
Ohana et al. [40]	U-Net	Euler	2×10^7	6:12	1.02
Ours	ViT	Euler	8.6×10^8	1:20	0.138
Ours	LNS $\div_{80}$	Euler	$3.1 \times 10^8 + 2.2 \times 10^8$	1:20	0.077
Ours	LDM $\div_{80}$	Euler	$3.1 \times 10^8 + 2.2 \times 10^8$	1:20	0.075
Ohana et al. [40]	FNO	RB	2×10^7	6:12	10+
Ohana et al. [40]	U-Net	RB	2×10^7	6:12	10+
Nguyen et al. [120]	PhysiX	RB	4.5×10^9	2:8	1.067
Wu et al. [121]	TANTE	RB	10^8	1:16	0.609
Mukhopadhyay et al. [122]	ViT + CSM	RB	10^8	10	0.140
Ours	ViT	RB	8.6×10^8	1:20	0.185
Ours	LNS $\div_{256}$	RB	$3.1 \times 10^8 + 2.2 \times 10^8$	1:20	0.197
Ours	LDM $\div_{256}$	RB	$3.1 \times 10^8 + 2.2 \times 10^8$	1:20	0.141
Ohana et al. [40]	FNO	TGC	2×10^7	6:12	3.55
Ohana et al. [40]	U-Net	TGC	2×10^7	6:12	7.14
Mukhopadhyay et al. [122]	ViT + CKM	TGC	10^8	10	0.527
Ours	LNS $\div_{48}$	TGC	$7.2 \times 10^8 + 2.2 \times 10^8$	1:10	0.302
Ours	LDM $\div_{48}$	TGC	$7.2 \times 10^8 + 2.2 \times 10^8$	1:10	0.296

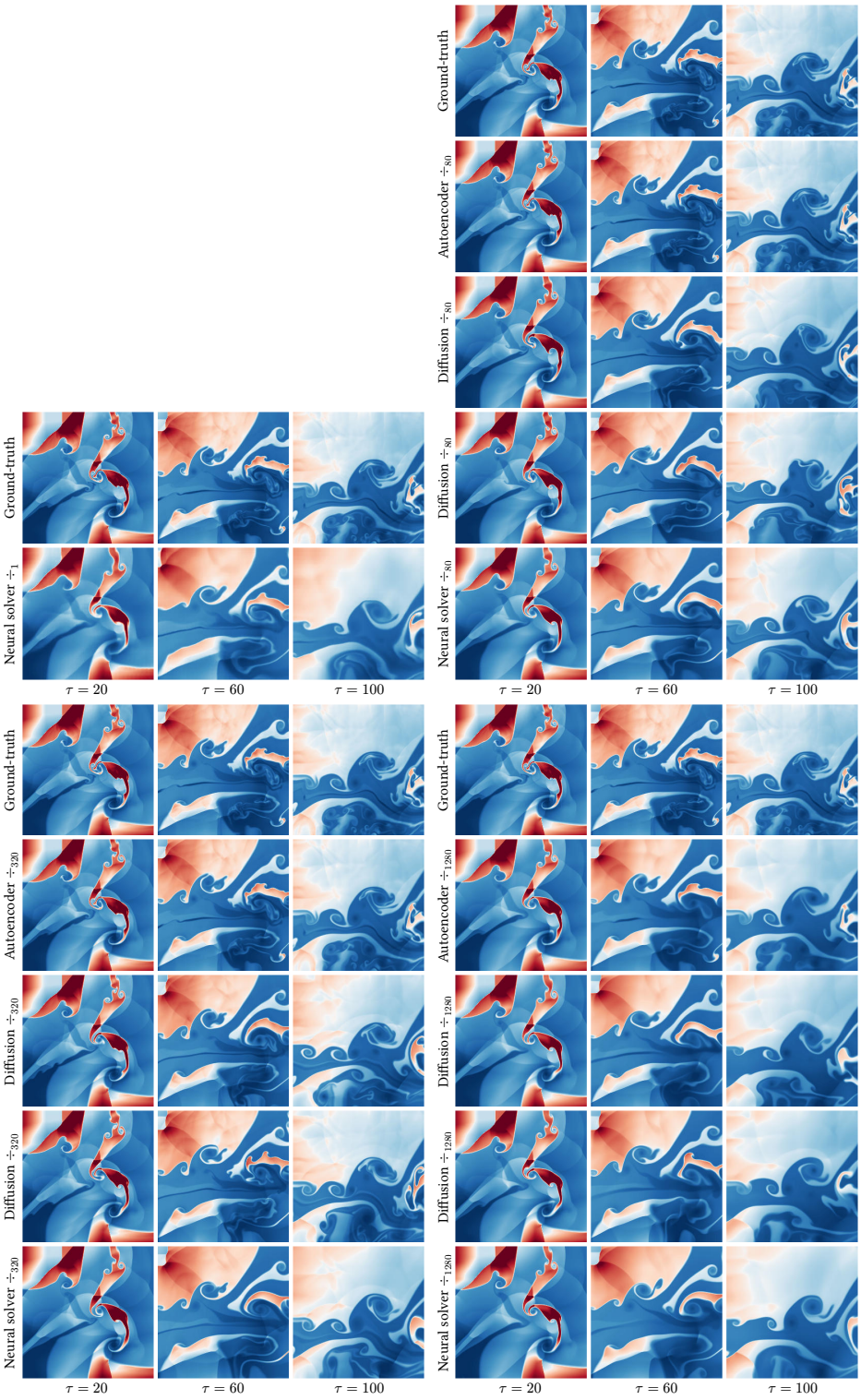


Figure 5.10. Examples of emulation at different compression rates (÷) for the Euler dataset. In this simulation, the system has open boundary conditions.

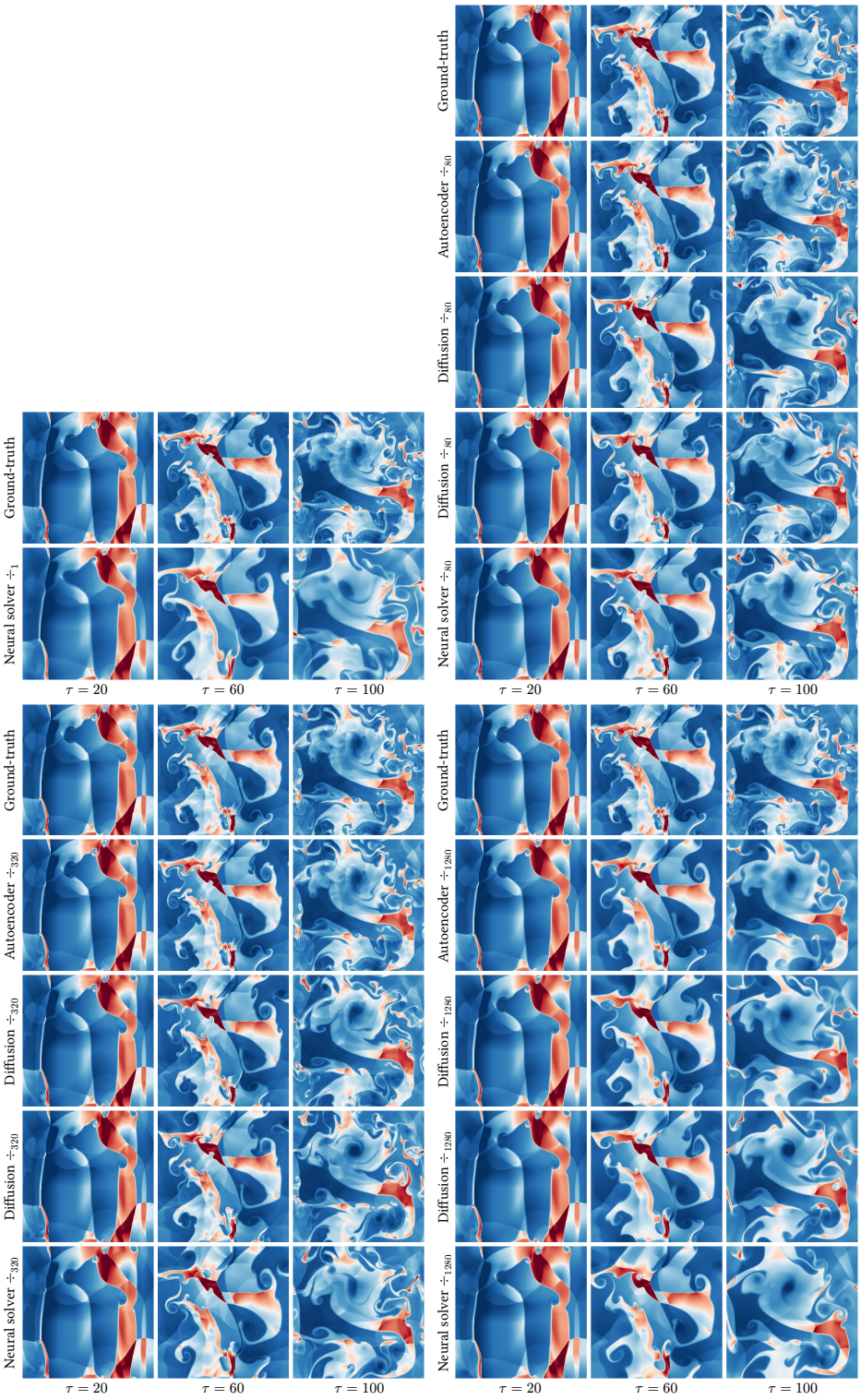


Figure 5.11. Examples of emulation at different compression rates (÷) for the Euler dataset. In this simulation, the system has periodic boundary conditions.

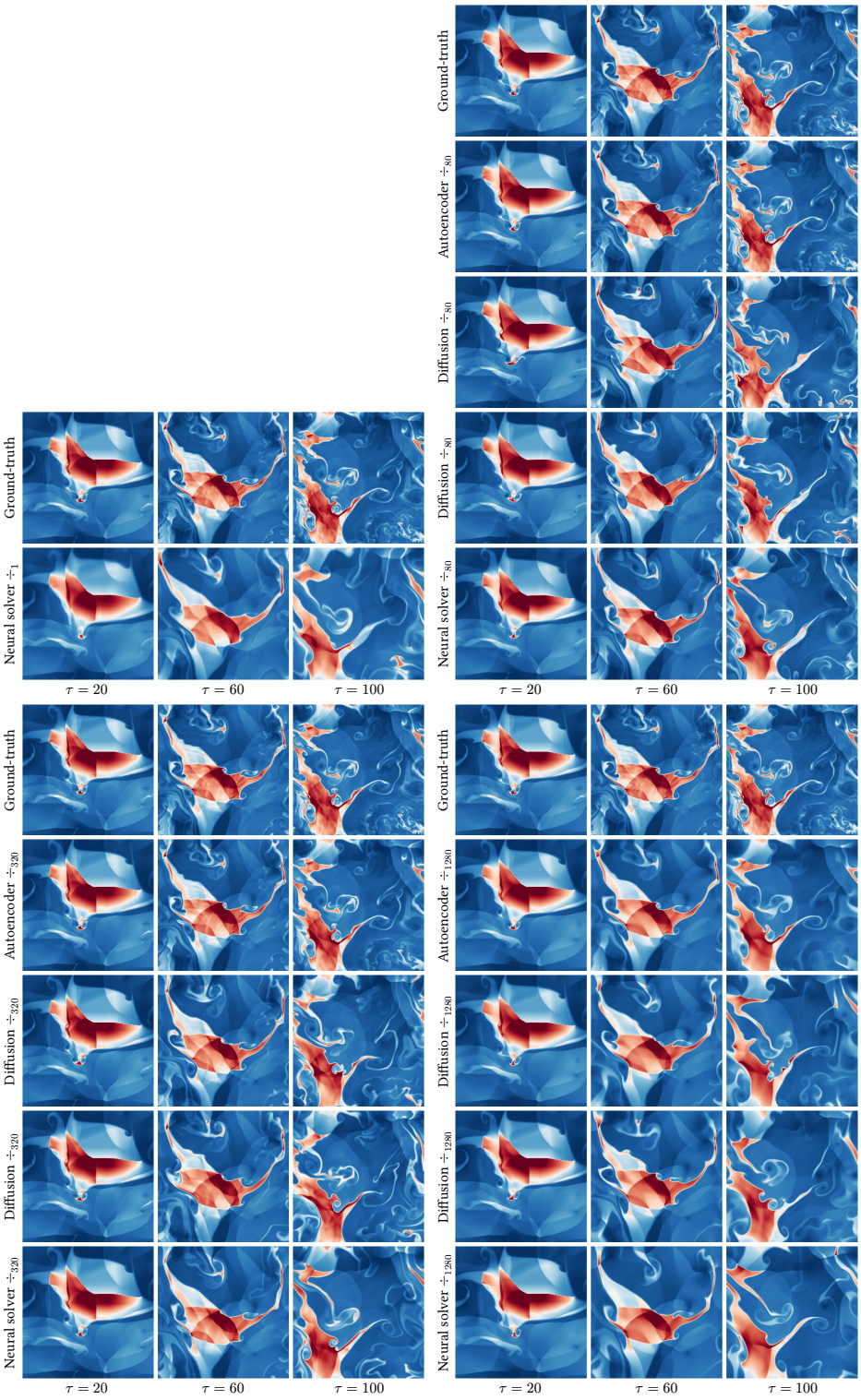


Figure 5.12. Examples of emulation at different compression rates (÷) for the Euler dataset. In this simulation, the system has periodic boundary conditions.

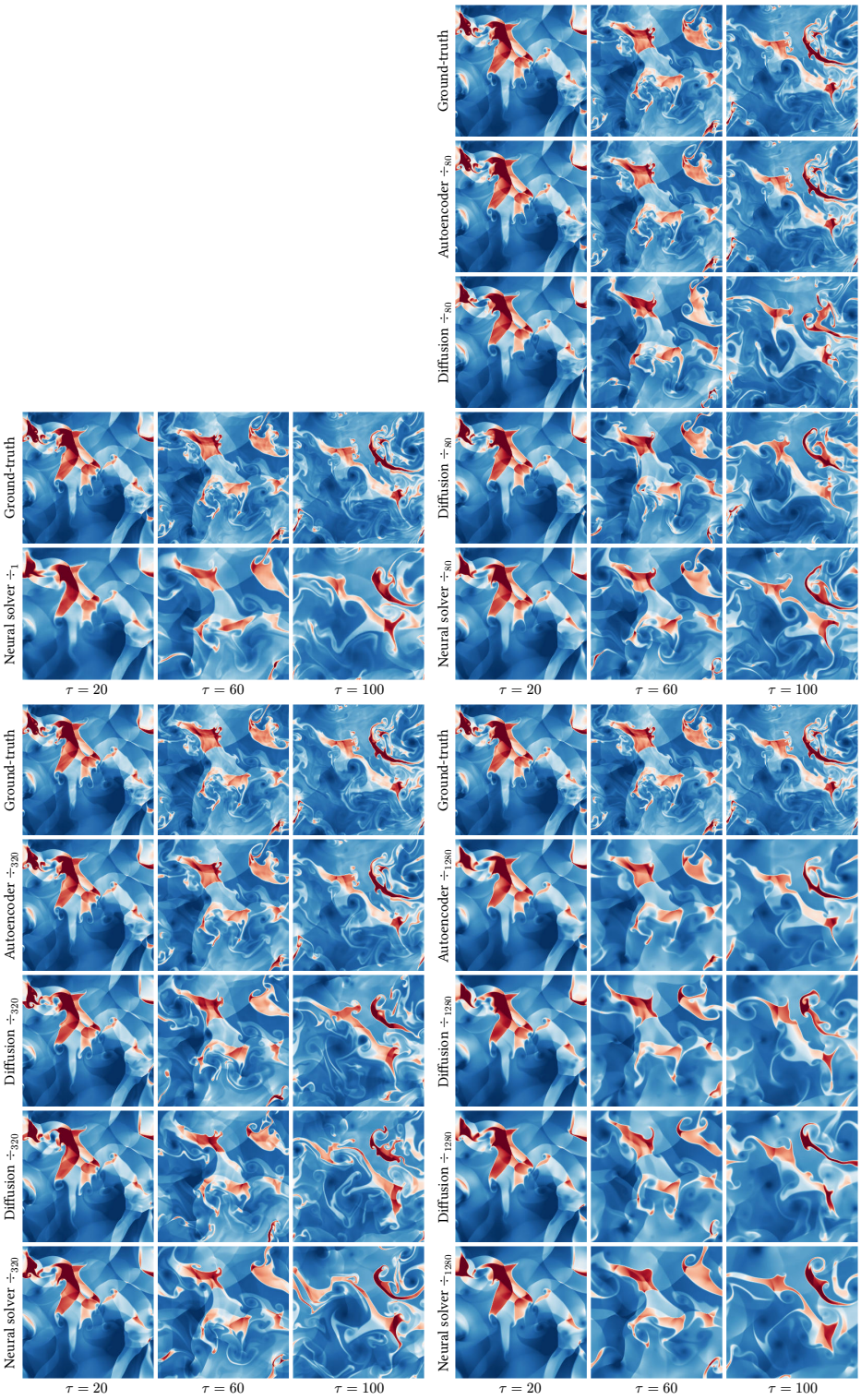


Figure 5.13. Examples of emulation at different compression rates (÷) for the Euler dataset. In this simulation, the system has periodic boundary conditions.

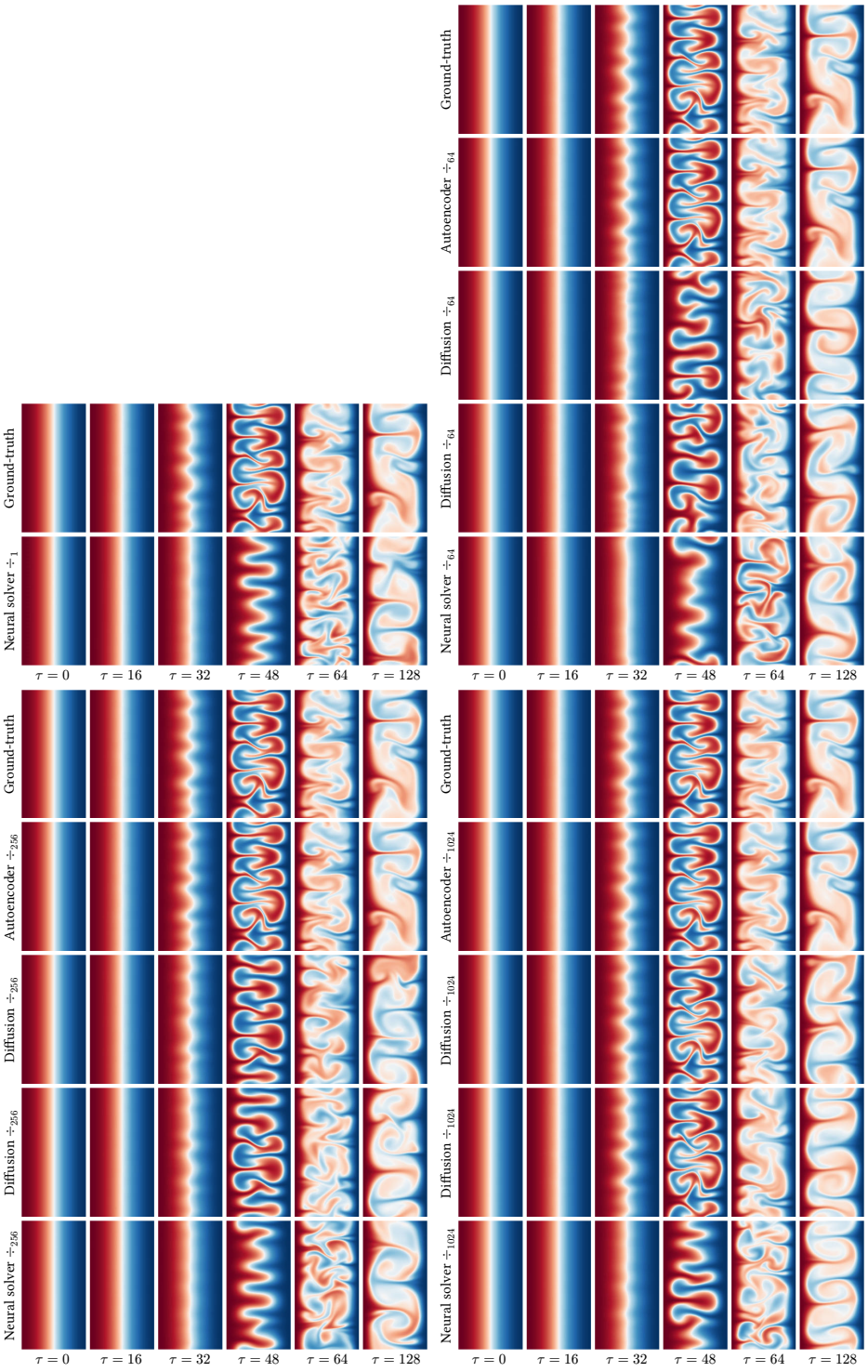


Figure 5.14. Examples of emulation at different compression rates (÷) for the Rayleigh-Bénard dataset. In this simulation, the fluid is in a low-turbulence regime ($\mathrm{Ra} = 10^6$).

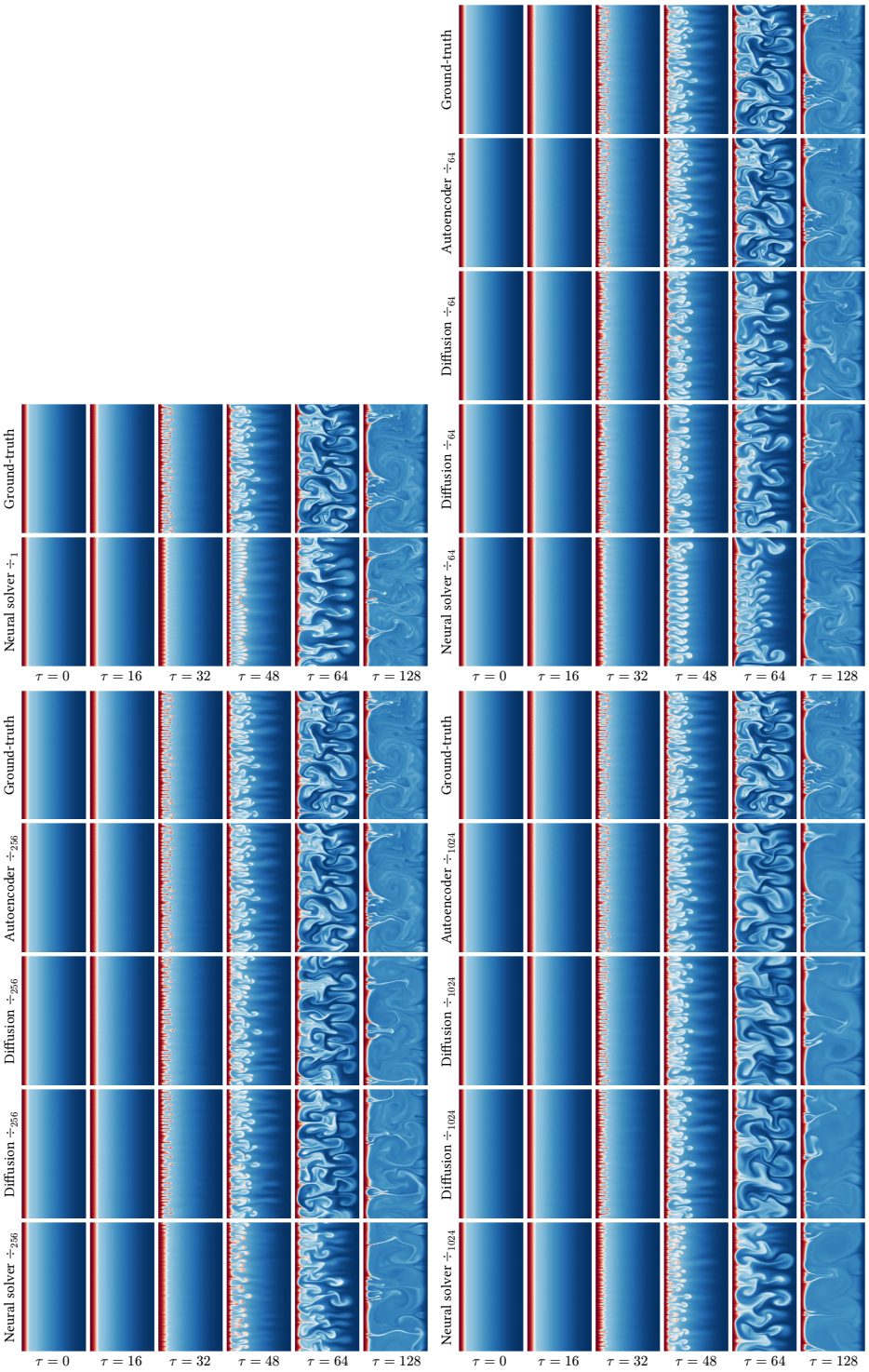


Figure 5.15. Examples of emulation at different compression rates (÷) for the Rayleigh-Bénard dataset. In this simulation, the fluid is in a high-turbulence regime ($\mathrm{Ra} = 10^8$).

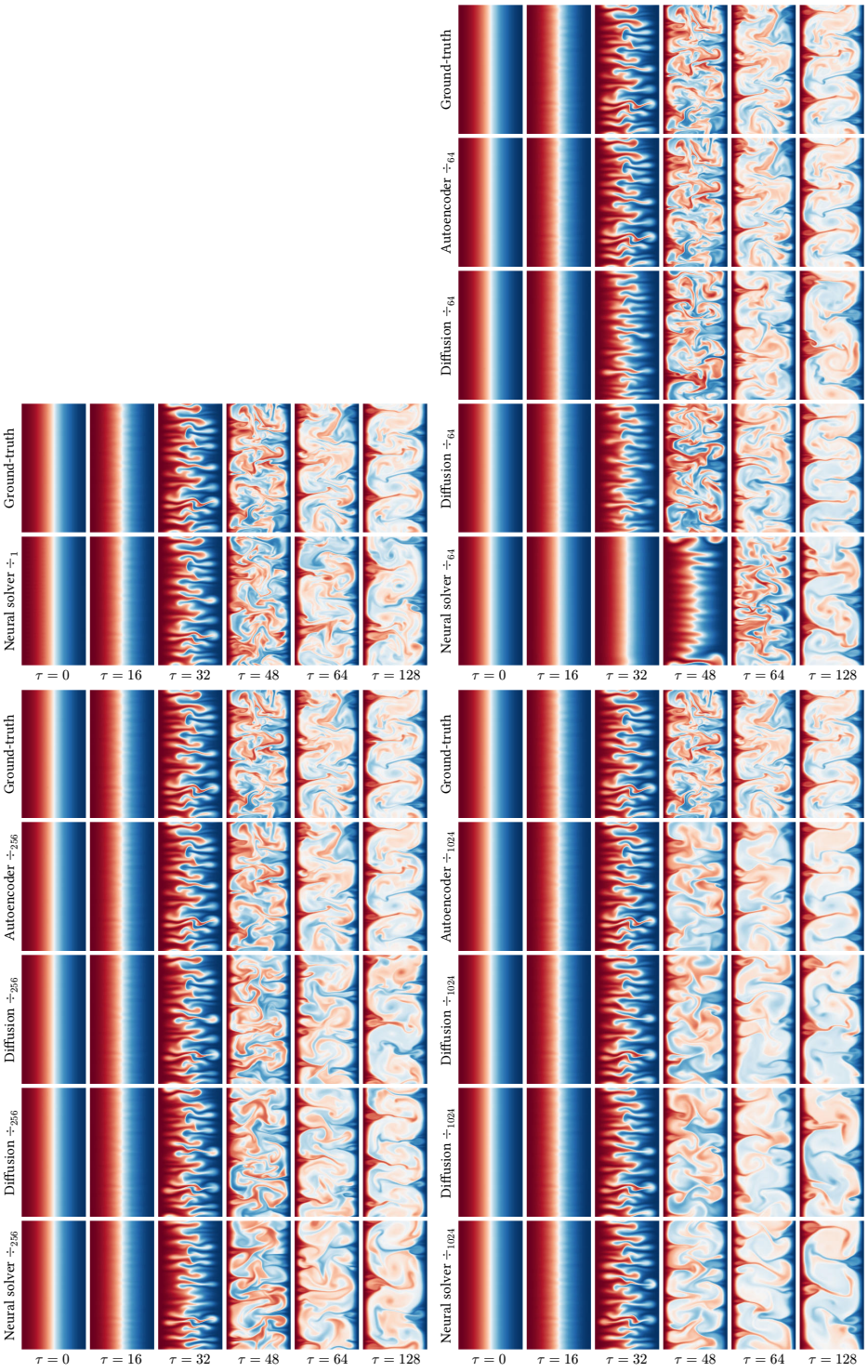


Figure 5.16. Examples of emulation at different compression rates (÷) for the Rayleigh-Bénard dataset. In this simulation, the fluid is in a low-turbulence regime ($\mathrm{Ra} = 10^6$).

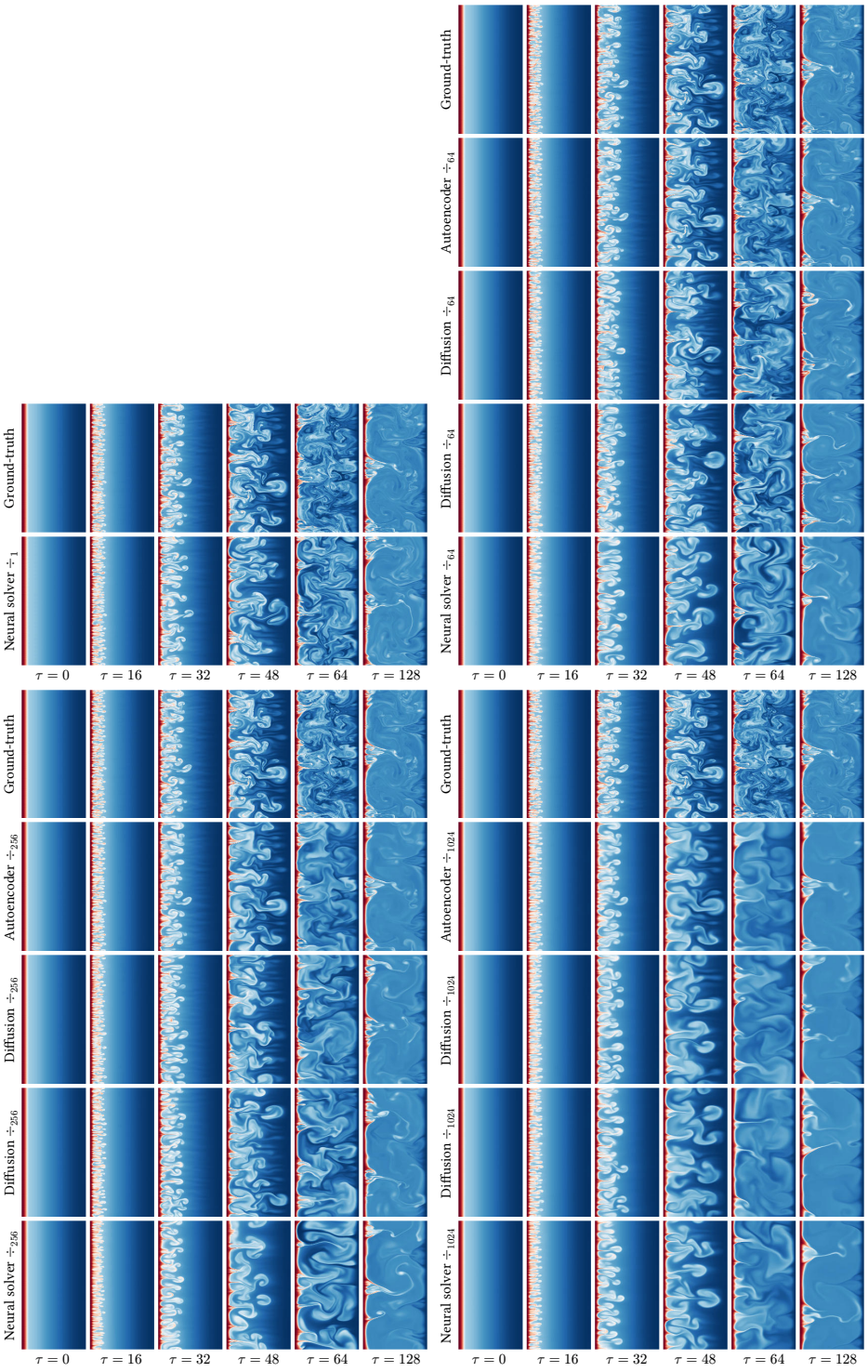


Figure 5.17. Examples of emulation at different compression rates (÷) for the Rayleigh-Bénard dataset. In this simulation, the fluid is in a high-turbulence regime ($\mathrm{Ra} = 10^8$).

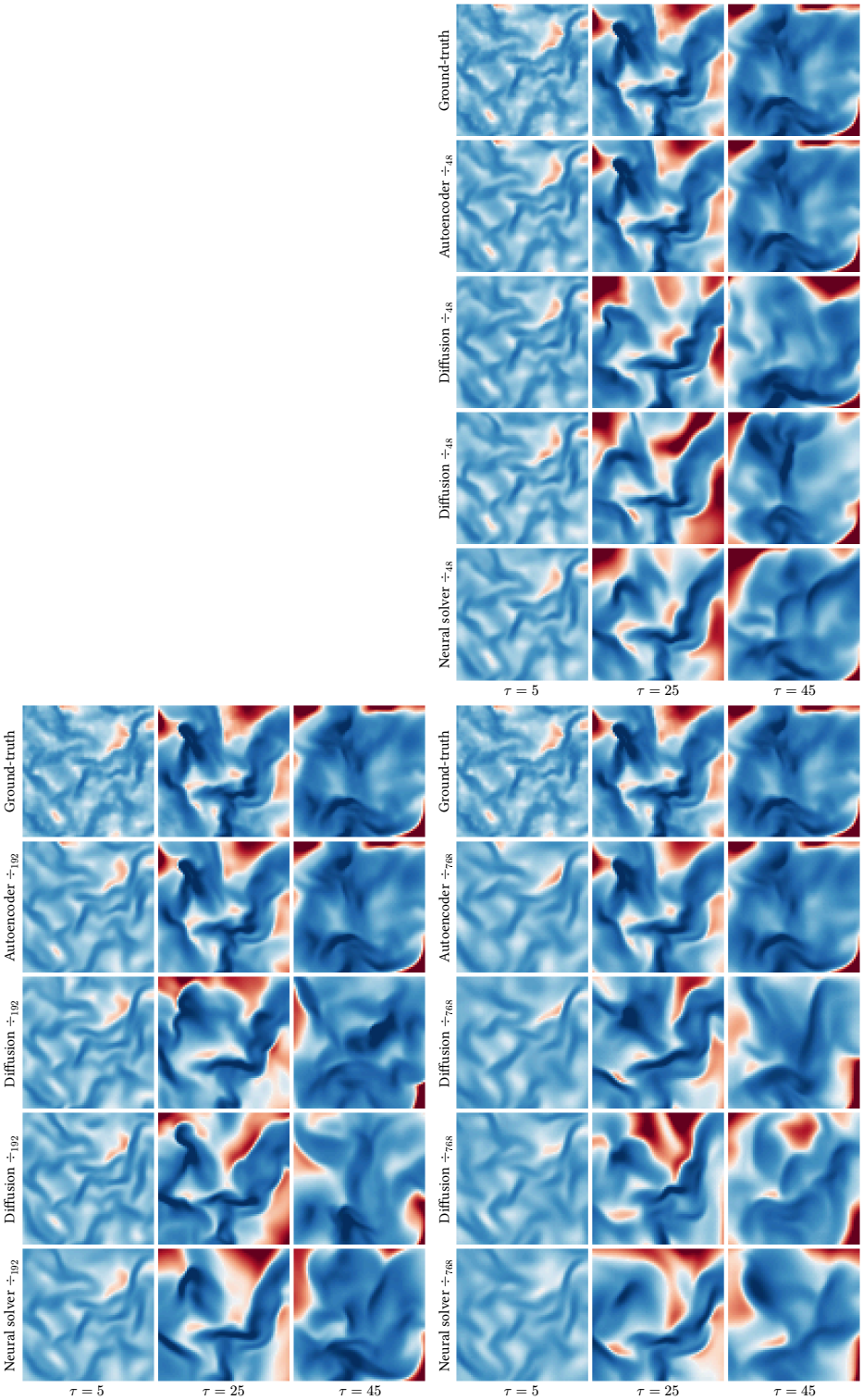


Figure 5.18. Examples of emulation at different compression rates (÷) for the TGC dataset. In this simulation, the initial density is low and the initial temperature is low ($\rho_0 = 0.445$, $T_0 = 10.0$).

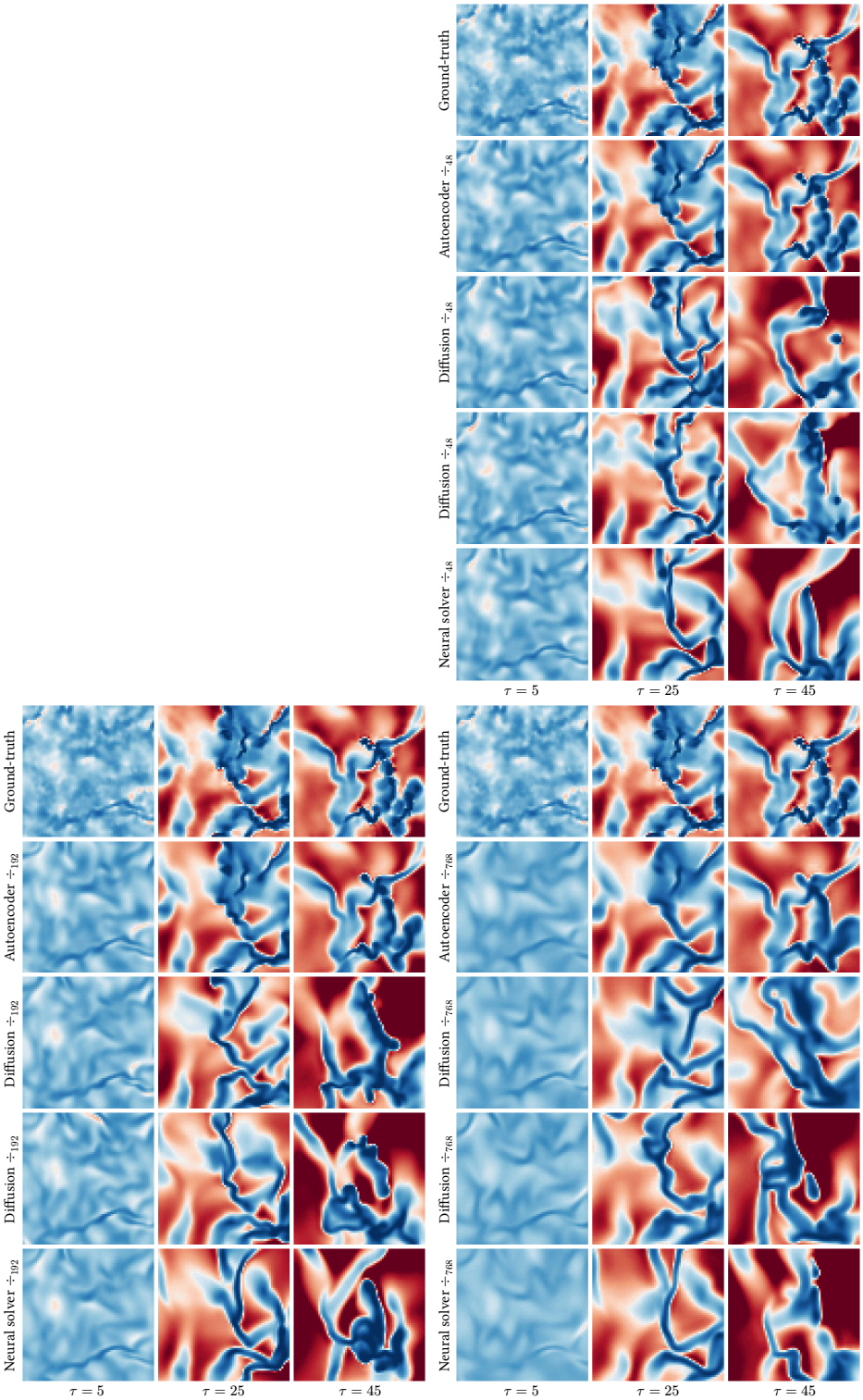


Figure 5.19. Examples of emulation at different compression rates (÷) for the TGC dataset. In this simulation, the initial density is medium and the initial temperature is high ($\rho_0 = 4.45$, $T_0 = 1000.0$).

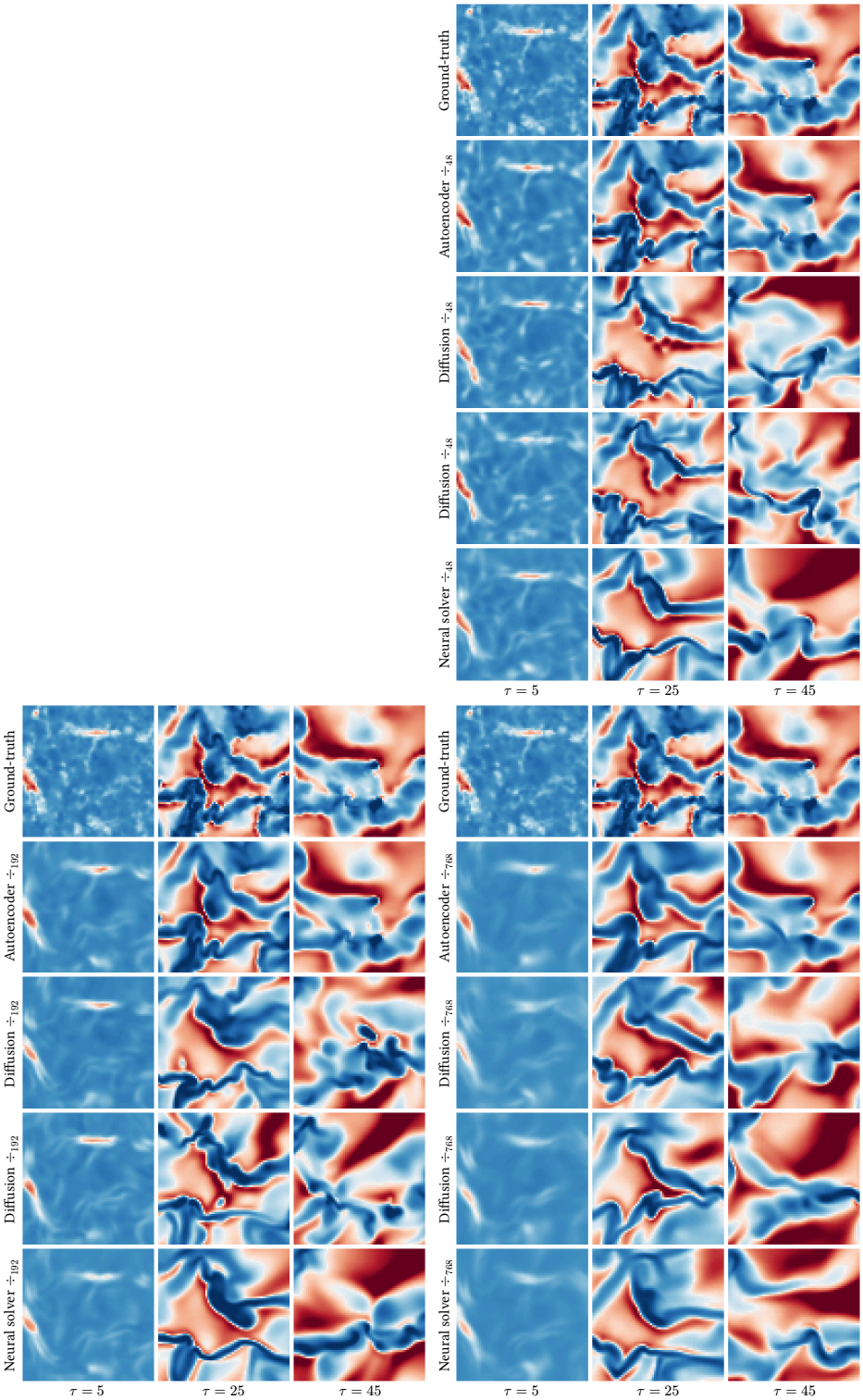


Figure 5.20. Examples of emulation at different compression rates (÷) for the TGC dataset. In this simulation, the initial density is high and the initial temperature is low ($\rho_0 = 44.5$, $T_0 = 10.0$).

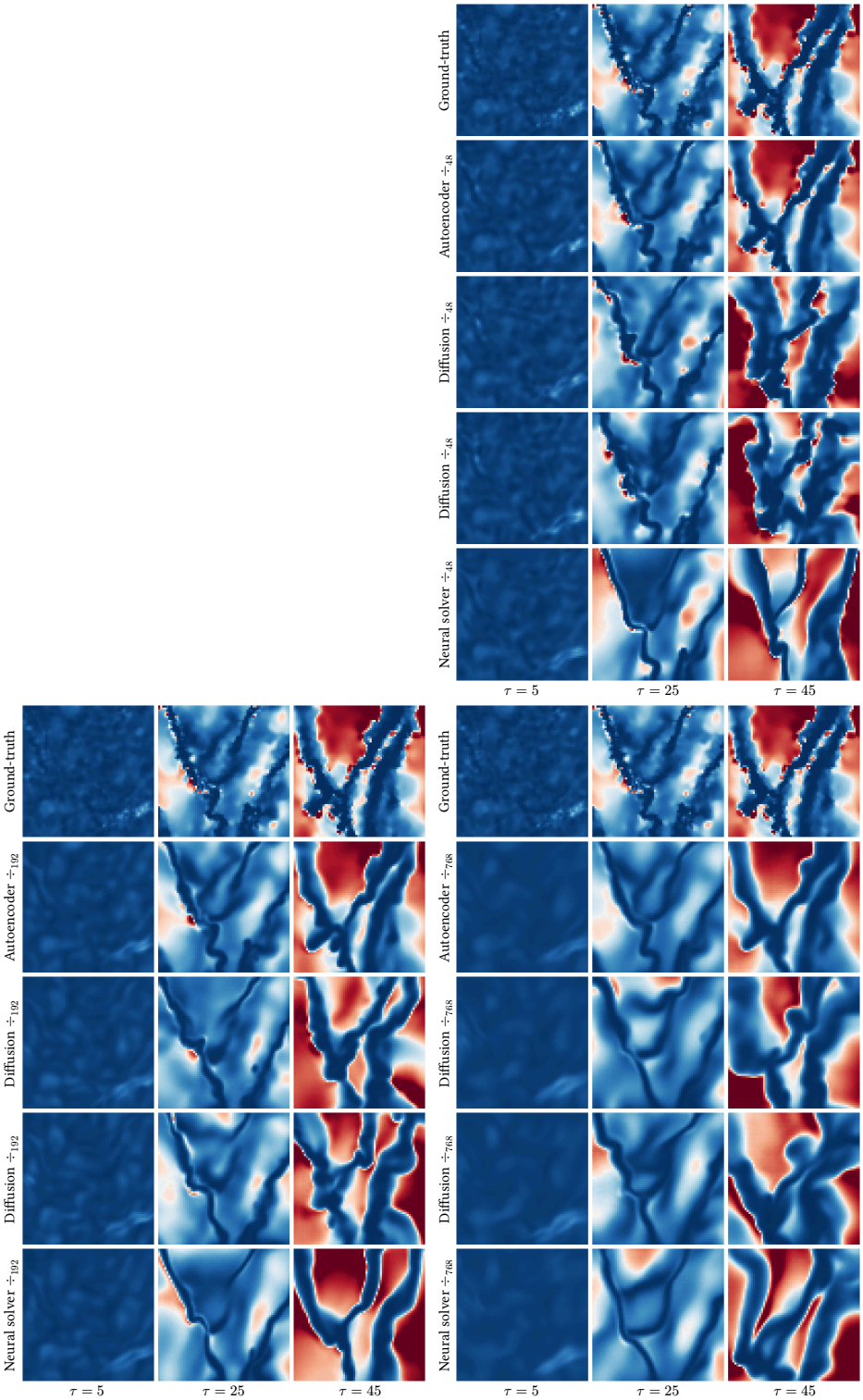


Figure 5.21. Examples of emulation at different compression rates (÷) for the TGC dataset. In this simulation, the initial density is high and the initial temperature is medium ($\rho_0 = 44.5$, $T_0 = 100.0$).

5.D Latent space analysis

In this section, we conduct a short analysis of the learned latent representations. We are notably interested in the separability of the latent representation with respect to different parameteres θ.

For our first experiment, we select a random initial state x^1 from the test split of the Euler dataset. We compute the initial state $z^1 = E_\psi(x^1)$ for the $\div_{80}$ autoencoder. For each heat capacity $\gamma \in \{1.2, 1.3, 1.4, 1.5, 1.6\}$, we generate one latent trajectory $z^{1:L}$ with the diffusion-based emulator. Afterwards we compute the Euclidean distance $\|z_a^i - z_b^i\|_2$ for each pair (γ_a, γ_b) of heat capacities. We report the results in Table 5.15 and represent the trajectories in Figure 5.22. As expected, trajectories with similar heat capacity γ are close to each others.

For our second experiment, we compute the latent representations $z^i = E_\psi(x^i) \in \mathbb{R}^{16\times4\times64}$ of the $\div_{64}$ autoencoder for randomly selected states x^i of the Rayleigh-Bénard dataset. We then train a small multi-layer perceptron (MLP) to predict the simulation parameters θ (Rayleigh and Prandtl numbers) from the latent state's central token $z^i[8, 2] \in \mathbb{R}^{64}$. We extract the activations of the MLP's last layer and visualize them with t-SNE [123] in Figure 5.23. We observe that t-SNE [123] continuously separates latent states with respect to their parameters θ, indicating that our autoencoders learn to distinguish between different physics. We further validate this result by computing the pairwise Bures-Wasserstein distances [124] between the distributions of central tokens $z^i[8, 2]$ for different Rayleigh and Prandtl numbers. The distances, reported in Tables 5.16 and 5.17, are anti-correlated with the similarity of simulation parameters θ.

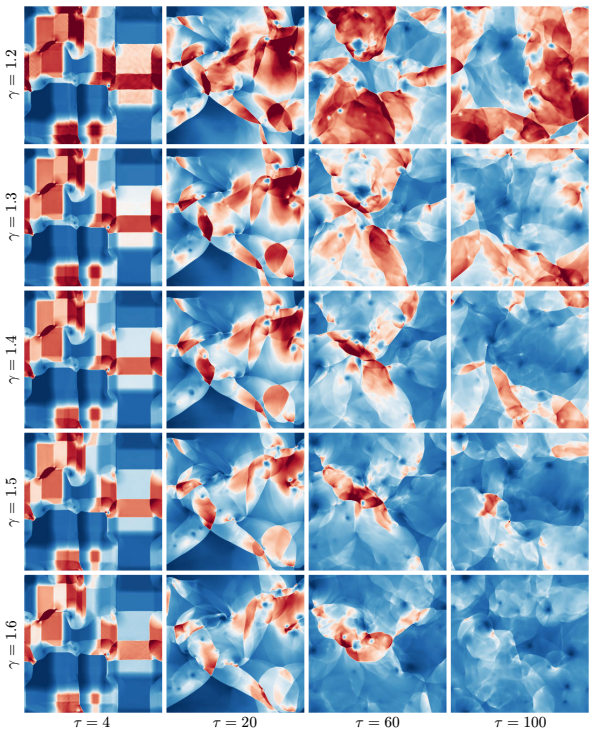


Figure 5.22. Example of emulated trajectories with different heat capacities $\gamma \in \{1.2, 1.3, 1.4, 1.5, 1.6\}$ but starting at the same initial state x^1 for the Euler dataset. The energy field is visualized instead of the density field to emphasize the differences.

Table 5.15. Euclidean distance matrix between emulated trajectories with different heat capacities $\gamma \in \{1.2, 1.3, 1.4, 1.5, 1.6\}$ but starting at the same initial state x^1 for the Euler dataset.

$\tau = 1$	1.2	1.3	1.4	1.5	1.6	$\tau = 20$	1.2	1.3	1.4	1.5	1.6
1.2	0.00	26.61	32.45	38.46	46.28	1.2	0.00	55.95	64.92	71.06	78.85
1.3	26.61	0.00	14.72	22.09	32.33	1.3	55.95	0.00	38.93	55.03	66.19
1.4	32.45	14.72	0.00	15.26	25.62	1.4	64.92	38.93	0.00	44.00	59.93
1.5	38.46	22.09	15.26	0.00	18.52	1.5	71.06	55.03	44.00	0.00	52.14
1.6	46.28	32.33	25.62	18.52	0.00	1.6	78.85	66.19	59.93	52.14	0.00
$\tau = 60$	1.2	1.3	1.4	1.5	1.6	$\tau = 100$	1.2	1.3	1.4	1.5	1.6
1.2	0.00	74.68	84.37	90.41	96.04	1.2	0.00	74.71	82.09	90.16	94.79
1.3	74.68	0.00	67.06	75.22	82.20	1.3	74.71	0.00	66.72	74.16	81.68
1.4	84.37	67.06	0.00	67.42	76.49	1.4	82.09	66.72	0.00	67.51	74.72
1.5	90.41	75.22	67.42	0.00	71.58	1.5	90.16	74.16	67.51	0.00	69.75
1.6	96.04	82.20	76.49	71.58	0.00	1.6	94.79	81.68	74.72	69.75	0.00

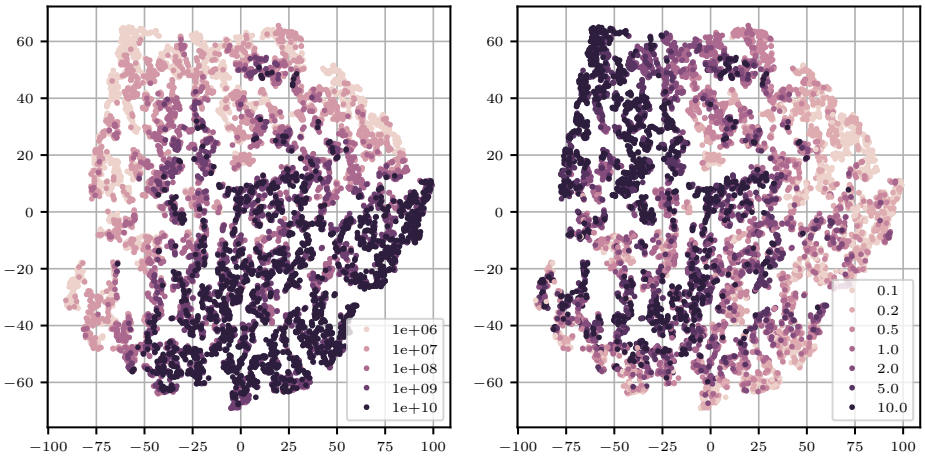


Figure 5.23. t-SNE [123] visualization of the latent states $z^i = E_\psi(x^i)$. The projections are colored with respect to their Rayleigh (left) and Prandtl (right) numbers.

Table 5.16. Bures-Wasserstein distance matrix between the distributions of latent states $z^i = E_\psi(x^i)$ with different Rayleigh numbers.

	10^6	10^7	10^8	10^9	10^{10}
10^6	0.000	1.045	1.708	2.279	2.489
10^7	1.045	0.000	0.965	1.537	1.794
10^8	1.708	0.965	0.000	0.915	1.180
10^9	2.279	1.537	0.915	0.000	0.714
10^{10}	2.489	1.794	1.180	0.714	0.000

Table 5.17. Bures-Wasserstein distance matrix between the distributions of latent states $z^i = E_\psi(x^i)$ with different Prandtl numbers.

	0.1	0.2	0.5	1.0	2.0	5.0	10.0
0.1	0.000	1.367	2.042	2.631	3.244	3.884	4.210
0.2	1.367	0.000	1.269	1.839	2.381	3.007	3.331
0.5	2.042	1.269	0.000	0.986	1.479	2.093	2.398
1.0	2.631	1.839	0.986	0.000	0.930	1.472	1.766
2.0	3.244	2.381	1.479	0.930	0.000	0.988	1.251
5.0	3.884	3.007	2.093	1.472	0.988	0.000	0.711
10.0	4.210	3.331	2.398	1.766	1.251	0.711	0.000

6 Latent Score-based Data Assimilation

Imagine a system on a rotating sphere that is 8000 miles wide, consists of different materials, different gases that have different properties (one of the most important of which, water, exists in different concentrations), heated by a nuclear reactor 98 million miles away. Then, just to make life interesting, this sphere is oriented such that, as it revolves around the nuclear reactor, it is heated differently at different locations at different times of the year. Then, someone is asked to watch the mixture of gases, a fluid only 20 miles deep, that covers an area of 250 million square miles, and to predict the state of that fluid at one point on the sphere two days from now. This is the problem weather forecasters face.

— Robert T. Ryan (1982)

Addendum

This chapter appeared previously as

> Gérôme Andry, Sacha Lewin, François Rozet, Omer Rochman, Victor Mangeleer, Matthias Pirlet, Elise Faulx, and Gilles Louppe. "Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2025.

Gérôme and Sacha implemented the methods, conducted the experiments, interpreted the results, and wrote most of the manuscript. François came up with the methods, supervised the codebase, and wrote the technical parts of the manuscript. Omer, Victor and Matthias contributed to the codebase and the visualization of the results. Elise provided valuable domain knowledge. Gilles supervised the project, suggested experiments, and participated in the writing.

For the version presented in this chapter, we have edited the structure and content of the manuscript to be consistent with the rest of this document. At the time of writing, the Appa project is still in progress and is not published within a peer-reviewed venue. Nevertheless, we wish to include this early version in this dissertation, as it is a logical continuation of our work.

Abstract

While deep learning has been transformative for numerical weather prediction, the assimilation step preceding forecasting that estimates the current atmospheric state from

observational data remains challenging. In this work, we introduce Appa, a score-based data assimilation model generating global atmospheric trajectories at 0.25° resolution and 1-hour intervals. Powered by a 565M latent diffusion model trained on ERA5, Appa can be conditioned on arbitrary observations to infer plausible trajectories, without retraining. Our probabilistic framework handles reanalysis, filtering, and forecasting as special cases. Results establish latent score-based data assimilation as a promising foundation for future global atmospheric modeling systems.

6.1 Introduction

Data assimilation combines observational data with physical models to estimate atmospheric states. Formally, let $x^{1:L} = (x^1, x^2, \dots, x^L) \in \mathbb{R}^{L\times V\times C}$ denote a trajectory of L atmospheric states, each represented as C physical fields over a mesh of V vertices. Let $p(x^1)$ be the initial state prior and $p(x^{i+1} \mid x^i)$ the transition dynamics. Observations $y^{1:L} \in \mathbb{R}^{M_1+\cdots+M_L}$ of the states $x^{1:L}$ follow an observation process $p(y^i \mid x^i)$, generally formulated as $y^i = \mathcal{H}^i(x^i) + \eta^i$, where the measurement functions $\mathcal{H}^i : \mathbb{R}^{V\times C} \mapsto \mathbb{R}^{M_i}$ can be non-linear and $\eta^i \in \mathbb{R}^{M_i}$ represents observational error that accounts for instrumental noise and systematic uncertainties. The goal of data assimilation is to infer plausible trajectories $x^{1:L}$ consistent with the observations $y^{1:L}$, that is, to estimate the trajectory posterior

$$p(x^{1:L} \mid y^{1:L}) \propto p(x^1) \prod_{i=1}^{L-1} p(x^{i+1} \mid x^i) \prod_{i=1}^{L} p(y^i \mid x^i). \tag{6.1}$$

While Eq. (6.1) defines the general assimilation problem, assimilation tasks correspond to specific choices of states x^i to infer and observations y^i to take into account. In this work, we focus on three special cases:

$$\textbf{Reanalysis} \quad p(x^{1:L} \mid y^{1:L}), \tag{6.2}$$

$$\textbf{Filtering} \quad p(x^{L} \mid y^{1:L}), \tag{6.3}$$

$$\textbf{Forecasting} \quad p(x^{K+1:L} \mid y^{1:K}). \tag{6.4}$$

Reanalysis aims to reconstruct full trajectories from historical observations. The primary purpose of reanalysis is to create datasets of historical data for the land, atmosphere, and oceans. These datasets enable scientists to better monitor and understand the climate, conduct surveys, and develop new weather models. Filtering, in contrast, only infers the posterior distribution of the last/current state x^L. This step is crucial to initialize global weather models in numerical weather prediction. Forecasting, as its name suggests, extends beyond the observed segment, producing posterior distributions over future states. The forecasting problem is often decomposed into first estimating the current state x^K from $y^{1:K}$, and then predicting its evolution.

Traditional assimilation methods like 4D-Var [3–7] or ensemble Kalman filters [8–12] are effective but rely on linearization and differentiation of the transition dynamics, which is very expensive at a global scale. In addition, 4D-Var provides point estimates instead of a posterior distribution [13]. Recent data-driven approaches [14–18] integrate deep learning into assimilation, but suffer from limited resolution, lack of uncertainty quantification, and require retraining for new observation configurations.

6.2 Appa

Appa combines score-based data assimilation [19–21] with latent diffusion-based physics emulation [22], scaled to the global atmospheric system at 0.25° resolution and 1-hour intervals. It consists of (1) a 340M encoder-decoder pair (E_ψ, D_ψ) that compresses

atmospheric states x^i into 530× smaller latent representations $z^i \sim \mathcal{N}(z^i \mid E_\psi(x^i), \sigma_z^2 I)$ and (2) a 225M diffusion transformer (DiT) [23] that generates windows $z^{i:i+W}$ of $W = 24$ consecutive latent states.

The encoder and decoder are trained to minimize the L_2 error between states x^i and their reconstruction $\hat{x}^i = D_\psi(z^i)$. The DiT $d_\phi(z_t^{i:i+W})$ is trained to estimate the denoising posterior mean $\mathbb{E}[z^{i:i+W} \mid z_t^{i:i+W}]$ for noisy latent states $z_t^{i:i+W} \sim \mathcal{N}(\alpha_t\, z^{i:i+W}, \sigma_t^2 I)$. Tweedie's formula [24–26]

$$\mathbb{E}\left[z^{i:i+W} \mid z_t^{i:i+W}\right] = \frac{z_t^{i:i+W} + \sigma_t^2 \nabla_{z_t^{i:i+W}} \log p(z_t^{i:i+W})}{\alpha_t} \tag{6.5}$$

then gives access to the prior score $\nabla_{z_t^{i:i+W}} \log p(z_t^{i:i+W})$ necessary to simulate the reverse diffusion process [27]. As in the SDA [19, 20] framework, the score over longer trajectories $z_t^{1:L}$ is approximated by composing the scores over windows $z_t^{i:i+W}$. Algorithm 10 generalizes SDA's composition algorithm by introducing a time stride $\Delta \geq 1$ between consecutive windows. Using a larger stride reduces the window overlap and, therefore, the number of network evaluations.

Sampling conditionally on observations To sample from the posterior $p(z^{1:L} \mid y^{1:L})$, we replace the prior score in the reverse diffusion process with the posterior score

$$\nabla_{z_t^{1:L}} \log p(z_t^{1:L} \mid y^{1:L}) = \nabla_{z_t^{1:L}} \log p(z_t^{1:L}) + \nabla_{z_t^{1:L}} \log p(y^{1:L} \mid z_t^{1:L}). \tag{6.6}$$

The prior score $\nabla_{z_t^{1:L}} \log p(z_t^{1:L})$ is obtained with the denoiser $d_\phi(z_t^{i:i+W})$ and Algorithm 10, while the likelihood score $\nabla_{z_t^{1:L}} \log p(y^{1:L} \mid z_t^{1:L})$ can be approximated [19, 28–30] without retraining under moderate assumptions on the observation process $p(y^i \mid x^i)$.

The key challenge is that the observation process $p(y^i \mid x^i)$ is defined in terms of atmospheric state x^i rather than latent state z^i. To address this issue, we approximate the mapping from z^i to y^i as the composition of the decoder D_ψ and measurement function $\mathcal{H}^i$. Formally, we assume

$$p(y^{1:L} \mid z^{1:L}) \approx \mathcal{N}\left(y^{1:L} \mid \mathcal{A}(z^{1:L}), \Sigma_y\right) \tag{6.7}$$

such that $\mathcal{A}(z^{1:L}) = \left(\mathcal{H}^1(D_\psi(z^1)) \ldots \mathcal{H}^L(D_\psi(z^L))\right)$ and Σ_y is the covariance of $\eta^{1:L}$. With this formulation, off-the-shelf posterior sampling algorithms [30] can be used to infer atmospheric trajectories conditionally on observational data. In this work, we adapt moment matching posterior sampling (MMPS) [29] to non-linear measurement functions $\mathcal{A}$ by estimating the covariance with the Jacobian A of $\mathcal{A}$, yielding the following perturbed likelihood approximation

$$p(y^{1:L} \mid z_t^{1:L}) \approx \mathcal{N}\left(y^{1:L} \mid \mathcal{A}\left(\mathbb{E}\left[z^{1:L} \mid z_t^{1:L}\right]\right), \Sigma_y + A\, \mathbb{V}\left[z^{1:L} \mid z_t^{1:L}\right] A^\top\right). \tag{6.8}$$

Algorithm 9 Training $d_\phi(z_t^{i:i+W})$

for $n = 1$ to N **do**
  $x^{1:L} \sim p(x^{1:L})$
  $i \sim \mathcal{U}(\{1, \ldots, L - W\})$
  $t \sim \mathcal{U}(0, 1), \varepsilon \sim \mathcal{N}(0, I)$
  for $j = i$ to $i + W$ **do**
    $z_j \leftarrow E_\psi(x_j)$
  $z_t^{i:i+W} \leftarrow \alpha_t\, z^{i:i+W} + \sigma_t\, \varepsilon$
  $\ell \leftarrow \left\| d_\phi(z_t^{i:i+W}) - z^{i:i+W} \right\|_2^2$
  $\phi \leftarrow \mathrm{SGD}(\phi, \nabla_\phi\, \ell)$

Algorithm 10 Composing $d_\phi(z_t^{i:i+W})$

function $d_\phi(z_t^{1:L})$
  $a \leftarrow {}^{(W-\Delta)}/_{2}$
  $b \leftarrow a + \Delta$
  $E_{1:a} \leftarrow d_\phi(z_t^{1:1+W})[:a]$
  for $n = 0$ to ${}^{(L-W)}/_{\Delta} + 1$ **do**
    $i \leftarrow 1 + n\Delta$
    $E_{i+a:i+b} \leftarrow d_\phi(z_t^{i:i+W})[a:b]$
  $E_{L-W+b:L} \leftarrow d_\phi(z_t^{L-W:L})[b:]$
  return $E_{1:L}$

Once the posterior score $\nabla_{z_t^{1:L}} \log p(z_t^{1:L} \mid y^{1:L})$ is available, we simulate the reverse diffusion process [27] to sample from $p(z^{1:L} \mid y^{1:L})$. Afterwards, the latent trajectories $z^{1:L}$ are mapped back to the atmospheric space via the decoder $\hat{x}^i = D_\psi(z^i)$.

6.3 Experiments

We train and evaluate Appa on the ERA5 reanalysis dataset [31]. We follow standard chronological splitting for evaluation: 1993–2019 for training, 2020 for validation, and 2021–2023 for testing. We consider 6 surface and 5 atmospheric fields across 13 pressure levels for a total of $C = 71$ physical fields. In the following, we evaluate the latent representation quality, performance across assimilation tasks, and compare against existing assimilation and forecasting methods.

Latent representation Despite the large (530×) compression factor, standardized reconstruction root-mean-squared error (RMSE) is mostly below 0.1, with slightly higher values for humidity and winds, and lower ones for surface and low-altitude fields, as shown in Figure 6.1. Power spectra of state reconstructions match the ground-truth closely at large scales, but slightly diverge for wavelengths smaller than 100 km, which is expected behavior from L_2 driven compression [29]. Since atmospheric energy concentrates at larger scales, the majority of energy is preserved. Overall, our autoencoder attains similar or better performance than prior neural compression methods of ERA5 [32].

Assimilation We evaluate Appa across four tasks: reanalysis, filtering, observational forecasting, and full-state forecasting. For the first three tasks, we assimilate synthetic ground station observations of all surface fields and simulated satellite scans of the 5 atmospheric fields. The station network consists of 11 000 real-world measurement locations [33] covering roughly 1 % of 0.25° grid points. Ground stations are sparse and globally distributed, while satellite orbital paths provide dense coverage within a limited spatio-temporal domain. Observations are modeled as Gaussian distributions centered on the ERA5 ground-truth, with noise levels of 1 % for ground stations and 10 % for satellites,

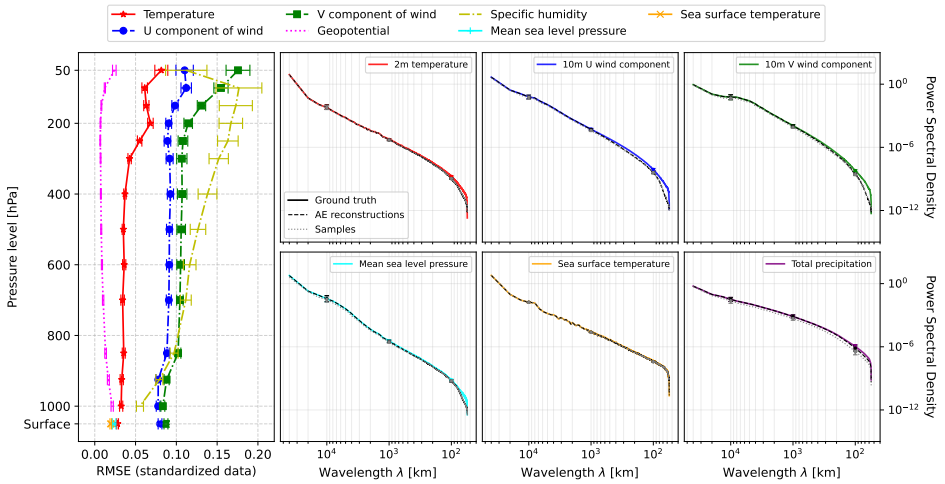


Figure 6.1. Standardized autoencoder reconstruction RMSE (left) and power spectra of Appa's prior samples (right) between January and December 2021. Lower-frequency fields (temperature, geopotential) are reconstructed more accurately than volatile fields (humidity, wind). Near-surface fields benefit from altitude-weighting. Power spectral density shows close alignment across wavelenghts, with deviations below 100 km.

relative to the observed field's respective standard deviation. Each assimilation is carried with 10 ensemble members.

Figure 6.2 summarizes Appa's quantitative performance. Further results can be found in Appendix 6.D. For reanalysis and filtering, conditioning on longer assimilation windows improves both skill and CRPS but gains saturate beyond 24 hours. Forecasting's skill decays gradually with lead time but remains significantly stronger than the persistence baseline. Observational forecasts, conditioned on the last 12 hours of a day-long assimilation, start at skill and CRPS levels comparable to the reanalysis plateau. Full-state forecasts, initialized from two complete states, start with lower errors but eventually converge to similar performance. The error growth lies between two baselines: steeper than IFS [34] but parallel to GraphDOP [35]. This suggests that the learned latent representation does not introduce artifacts that strongly impede the learning of the dynamics, supporting Rozet et al. [22]'s findings.

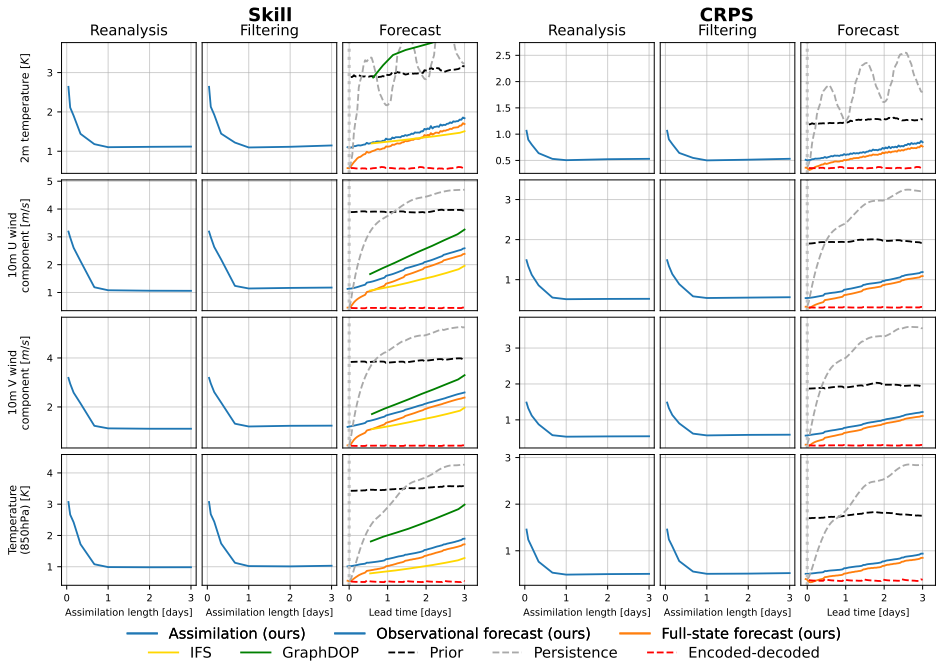


Figure 6.2. Skill and continuous ranked probability score (CRPS) of representative fields in assimilation tasks over January 2023. Reanalysis metrics are averaged over assimilation windows $x^{1:L}$, while filtering only considers the last assimilated state x^L. Both improve with longer assimilation windows, but eventually stagnate. Forecasts gradually lose skill over lead time. IFS [34] and GraphDOP [35] are reported for reference.

6.4 Discussion

Summary We introduce Appa, a latent score-based data assimilation model that produces global atmospheric trajectories efficiently by operating in a compressed latent space. Appa can be conditioned on arbitrary observations without retraining, providing access to the full posterior distribution of consistent trajectories. Our results demonstrate that Appa flexibly handles reanalysis, filtering, and forecasting within a unified probabilistic framework, while remaining competitive with baselines.

Limitations and future work While Appa demonstrates strong assimilation and forecasting capabilities, it remains a proof of concept that isn't ready yet for operational applications. We foresee several possible improvements. First, we should move from simplified synthetic observations to realistic measurements, such as satellite radiances. Improving physical consistency is also critical, especially for forecasting tasks where Appa falls short of standard baselines such as IFS [34]. Replacing guidance-based posterior sampling with explicit conditioning on past states, as in GenCast [36] and LoLA [29], could improve both accuracy and efficiency of forecasting. In terms of statistical assessment, the calibration of posterior distributions deserves further validation. The approximations present in our method, notably while estimating the prior and likelihood scores, introduce errors which are hard to quantify without proper statistical validation. Computational efficiency of reanalysis remains a challenge as well, as conditioning with respect to atmospheric observations requires decoding latent states repeatedly. Projecting observations into latent space could mitigate this bottleneck [37]. Finally, our comparison to other models is still preliminary due to the discrepancies in experimental configurations. Fair evaluation against IFS [34], GraphDOP [35] and other models [36, 38] would help position Appa within the spectrum of global atmospheric models.

Acknowledgments

Gérôme Andry, François Rozet, Sacha Lewin, and Elise Faulx are research fellows of the F.R.S.-FNRS (Belgium) and acknowledge its financial support. Omer Rochman acknowledges the financial support of the Walloon Region under grant no. 2.0102.35 (ARIAC by Digital Wallonia 4.AI). Victor Mangeleer is a research fellow part of the Multiple THreats on Ocean health (MiTHO) project and acknowledges funding from the European Space Agency (ESA).

The present research benefited from computational resources made available on Lucia, the Tier-1 supercomputer of the Walloon Region, infrastructure funded by the Walloon Region under the grant no. 1910247. The computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under the grant no. 2.5020.11 and by the Walloon Region.

References

[1] Robert T. Ryan. "The Weather Is Changing ... or Meteorologists and Broadcasters, the Twain Meet". In *Bulletin of the American Meteorological Society* 63.3 (1982).

[2] Gérôme Andry et al. "Appa: Bending Weather Dynamics with Latent Diffusion Models for Global Data Assimilation". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2025.

[3] A. C. Lorenc. "Analysis methods for numerical weather prediction". In *Quarterly Journal of the Royal Meteorological Society* 112.474 (1986).

[4] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[5] Yannick Trémolet. "Accounting for an imperfect model in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).

[6] Yannick Trémolet. "Model-error estimation in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 133.626 (2007).

[7] Mike Fisher et al. "Weak-constraint and long window 4DVAR". Tech. rep. ECMWF, 2011.

[8] Geir Evensen. "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In *Journal of Geophysical Research: Oceans* 99.C5 (1994).

[9] Craig H. Bishop, Brian J. Etherton, and Sharanya J. Majumdar. "Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects". In *Monthly Weather Review* 129.3 (2001).

[10] Jeffrey L. Anderson. "An adaptive covariance inflation error correction algorithm for ensemble filters". In *Tellus A: Dynamic Meteorology and Oceanography* 59.2 (2007).

[11] Brian R. Hunt, Eric J. Kostelich, and Istvan Szunyogh. "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter". In *Physica D: Nonlinear Phenomena*. Data Assimilation 230.1 (2007).

[12] Geir Evensen. "Data Assimilation: The Ensemble Kalman Filter". Springer, 2009.

[13] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[14] Sibo Cheng et al. "Machine Learning With Data Assimilation and Uncertainty Quantification for Dynamical Systems: A Review". In *IEEE/CAA Journal of Automatica Sinica* 10.6 (2023).

[15] Langwen Huang et al. "DiffDA: a Diffusion model for weather-scale Data Assimilation". In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[16] Xiaoze Xu et al. "FuXi-DA: a generalized deep learning data assimilation framework for assimilating satellite observations". In *npj Climate and Atmospheric Science* 8.1 (2025).

[17] Ronan Fablet et al. "Joint Interpolation and Representation Learning for Irregularly Sampled Satellite-Derived Geophysical Fields". In *Frontiers in Applied Mathematics and Statistics* 7 (2021).

[18] Marcin Andrychowicz et al. "Deep Learning for Day Forecasts from Sparse Observations". 2023.

[19] François Rozet and Gilles Louppe. "Score-based Data Assimilation". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[20] François Rozet and Gilles Louppe. "Score-based Data Assimilation for a Two-Layer Quasi-Geostrophic Model". In *Machine Learning and the Physical Sciences Workshop (NeurIPS)*. 2023.

[21] Jonathan Schmidt et al. "A Generative Framework for Probabilistic, Spatiotemporally Coherent Downscaling of Climate Simulation". In *npj Climate and Atmospheric Science* 8.1 (2025).

[22] François Rozet et al. "Lost in Latent Space: An Empirical Study of Latent Diffusion Models for Physics Emulation". In *Advances in Neural Information Processing Systems*. Vol. 38. 2025.

[23] William Peebles and Saining Xie. "Scalable Diffusion Models with Transformers". In *IEEE/CVF International Conference on Computer Vision*. 2023.

[24] M. C. K. Tweedie. "Functions of a statistical variate with given means, with special reference to Laplacian distributions". In *Mathematical Proceedings of the Cambridge Philosophical Society* (1947).

[25] Bradley Efron. "Tweedie's Formula and Selection Bias". In *Journal of the American Statistical Association* (2011).

[26] Kwanyoung Kim and Jong Chul Ye. "Noise2Score: Tweedie's Approach to Self-Supervised Image Denoising without Clean Images". In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[27] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In *International Conference on Learning Representations*. 2021.

[28] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In *International Conference on Learning Representations*. 2023.

[29] François Rozet et al. "Learning Diffusion Priors from Observations by Expectation Maximization". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[30] Giannis Daras et al. "A Survey on Diffusion Models for Inverse Problems". 2024.

[31] Hans Hersbach et al. "The ERA5 global reanalysis". In *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020).

[32] Piotr Mirowski et al. "Neural Compression of Atmospheric States". 2024.

[33] NOAA National Centers of Environmental Information. "Global Surface Summary of the Day - GSOD". 1999.

[34] ECMWF. "IFS documentation CY47R1 - part II: Data assimilation". In *IFS Documentation CY47R1*. ECMWF, 2020.

[35] Mihai Alexe et al. "GraphDOP: Towards skilful data-driven medium-range weather forecasts learnt and initialised directly from observations". 2024.

[36] Ilan Price et al. "Probabilistic weather forecasting with machine learning". In *Nature* 637.8044 (2025).

[37] Ron Raphaeli, Sean Man, and Michael Elad. "SILO: Solving Inverse Problems with Latent Operators". 2025.

[38] Remi Lam et al. "Learning skillful medium-range global weather forecasting". In *Science* 382.6677 (2023).

[39] Stephan Rasp et al. “WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models”. In *Journal of Advances in Modeling Earth Systems* 16.6 (2024).

[40] Nikhil Vyas et al. “SOAP: Improving and Stabilizing Shampoo using Adam for Language Modeling”. In *International Conference on Learning Representations*. 2025.

[41] Yang Song and Stefano Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution”. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[42] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In *International Conference on Learning Representations*. 2015.

[43] V. Fortin et al. “Why Should Ensemble Spread Match the RMSE of the Ensemble Mean?” In *Journal of Hydrometeorology* 15.4 (2014).

[44] Hans Hersbach. “Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems”. In *Weather and Forecasting* 15.5 (2000).

[45] Tilmann Gneiting and Adrian E Raftery. “Strictly Proper Scoring Rules, Prediction, and Estimation”. In *Journal of the American Statistical Association* 102.477 (2007).

6.A Data

ERA5 is a global deterministic reanalysis dataset from ECMWF that provides high-resolution (0.25°) hourly estimates of atmospheric, land, and oceanic fields from 1959 onward [31]. It assimilates observations into a numerical weather prediction model using 4D-Var data assimilation.

In this work, we use a subset of ERA5 data, defined on a 0.25° equiangular grid with 13 pressure levels: 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa. Following Lam et al. [38], we restrict the temporal coverage of the dataset to the 1993–2021 period, with data split into training (1993–2019), validation (2020), and testing (2021–2023). ERA5 data was downloaded from the WeatherBench2 [39] platform, where Google has made it available via Google Cloud Storage.

Table 6.1 lists the selected fields. Some fields serve as both input and output (predicted) features, while others provide contextual information as inputs but are not predicted.

Table 6.1. List of selected fields.

Type	Name	Role
Atmospheric	Temperature	Input / Output
Atmospheric	U-Wind Component	Input / Output
Atmospheric	V-Wind Component	Input / Output
Atmospheric	Geopotential	Input / Output
Atmospheric	Specific Humidity	Input / Output
Surface	2m Temperature	Input / Output
Surface	10m U-Wind Component	Input / Output
Surface	10m V-Wind Component	Input / Output
Surface	Mean Sea Level Pressure	Input / Output
Surface	Sea Surface Temperature	Input / Output
Surface	Total Precipitation	Input / Output
Time	Local time of day	Input (diffusion)
Time	Year progress	Input (diffusion)

6.A.1 Data processing

Standardization Although the dynamics across the atmospheric column are correlated, each pressure level exhibits distinct statistical behavior. Thus, we compute the mean and standard deviation separately for each field and at each pressure level, on the whole training dataset. We use these statistics to standardize our entire dataset and to rescale the output of Appa.

Missing values As sea surface temperature is undefined over land (NaN values), we replace these with zeros after standardization as a neutral placeholder.

6.B Technical details

This section provides further technical details for training and inference. Our code will be made available with full reproducibility steps for both training and evaluation.

6.B.1 Architectures

We adapted architectures from Rozet et al. [22] for both autoencoder and latent diffusion model. The encoder and decoder are fully convolutional neural networks and the diffusion model is adapted from a diffusion transformer (DiT) [23].

Autoencoder The autoencoder compresses atmospheric states from the high-dimensional N320 grid (721×1440 pixels with 71 channels) into a compact latent space (23×47 pixels with 128 channels) via progressive spatial downsampling and channel expansion. To accommodate any spatial compression factor, the input is padded to the nearest compatible grid size. We apply circular padding along longitude to respect periodicity, and constant zero padding along latitude to handle polar boundaries. The encoder-decoder pair is trained with a latitude-altitude-weighted mean squared error loss, following Lam et al. [38].

Table 6.2. Autoencoder training configuration

Parameter	Value
Loss function	latitude-altitude-weighted mean squared error
Latent noise	$\sigma_z = 0.01$ for regularization
Optimizer	SOAP [40] with initial learning rate 3×10^{-5} and linear decay
Batch size	64 samples per optimizer step
Training duration	95000 optimizer steps (2 days)
Hardware	64 A100 (40GB) GPUs

Latent denoiser The denoiser is a DiT that operates on $W = 24$ consecutive latent states. We patchify the latent sequence by a factor of 2 along the temporal axis and flatten the spatial dimensions, yielding a total of $23 \times 47 \times {}^{24}/_{2} = 12\,972$ tokens, each with 256 channels. During training and inference, we adopt a variance exploding noise process [41] for which $\alpha_t = 1$ and $\sigma_t \in [\sigma_{\min}, \sigma_{\max}]$.

Table 6.3. Denoiser training configuration

Parameter	Value
Loss	Denoising score matching
Noise range	$\sigma_{\min} = 10^{-3}$, $\sigma_{\max} = 10^{3}$
Optimizer	Adam [42] with initial learning rate 10^{-4}
Batch size	256 samples per optimizer step
Training duration	125000 optimizer steps (5 days)
Hardware	64 A100 (40GB) GPUs

6.B.2 Assimilation

Forecasting Appa is trained to generate state windows of $W = 24$ hours. To perform forecasting, we split the window in two parts: the conditioned states and the predicted

states. We either use the result of a reanalysis (observational forecasting) or encoded ground-truth atmospheric states (full-state forecasting) as condition, to infer the first window. For forecast of more than 24 hours, we perform autoregressive rollout. We use the last latent states of the previous window to condition the next one. The number of conditioning states can be tuned to carry more or less information from the past states.

Evaluation In Figure 6.2, we borrow IFS [34] and GraphDOP [35] metrics from Alexe et al. [35]. However, a direct and fair comparison is difficult due to the lack of experimental details. Alexe et al. [35] report metrics for 6 different fields over January 2023, but do not mention the exact dates or ensemble size. We selected the first 8 days of January 2023 at midnight as starting time. We compute metrics for 10 ensemble members.

6.C Evaluation metrics

We follow conventional metrics for assimilation and forecasting performance. For a fair comparison with the literature, evaluation is performed using WeatherBench2 [39]. For assimilation, we average performance over the time steps.

Skill The skill [36, 43] of an ensemble of K particles v_k is defined as the RMSE of the ensemble mean compared to the ground-truth u

$$\text{Skill} = \sqrt{\left\langle \left(u - \frac{1}{K} \sum_{k=1}^{K} v_k \right)^2 \right\rangle} \tag{6.9}$$

where $\langle \cdot \rangle$ denotes the spatial mean operator.

Spread The spread [36, 43] is defined as the ensemble standard deviation

$$\text{Spread} = \sqrt{\left\langle \frac{1}{K-1} \sum_{j=1}^{K} \left(v_j - \frac{1}{K} \sum_{k=1}^{K} v_k \right)^2 \right\rangle} \tag{6.10}$$

Spread-Skill ratio Under these definitions and the assumption of a perfect forecast where ensemble particles are exchangeable, Fortin et al. [43] show that

$$\text{Skill} = \sqrt{\frac{K+1}{K}} \text{Spread} \tag{6.11}$$

This motivates the use of the (corrected for ensemble size) spread-skill ratio as a metric. Intuitively, if the ratio is smaller than one, the ensemble is biased or under-dispersed. If the ratio is larger than one, the ensemble is over-dispersed. It should be noted, however, that a spread-skill ratio of 1 is a necessary but not sufficient condition for a perfect forecast.

CRPS The continuous ranked probability score (CRPS) [44, 45] is defined as

$$\text{CRPS} = \left\langle \frac{1}{K} \sum_{k=1}^{K} |u - v_k| - \frac{1}{2K(K-1)} \sum_{j=1}^{K} \sum_{k=1}^{K} |v_j - v_k| \right\rangle \tag{6.12}$$

The first term penalizes the distance of the particles to the ground-truth while the second term encourages variations within the ensemble. Minimizing the CRPS (CRPS $= 0$) means that the 1-d marginals of the ensemble are calibrated.

6.D Additional results

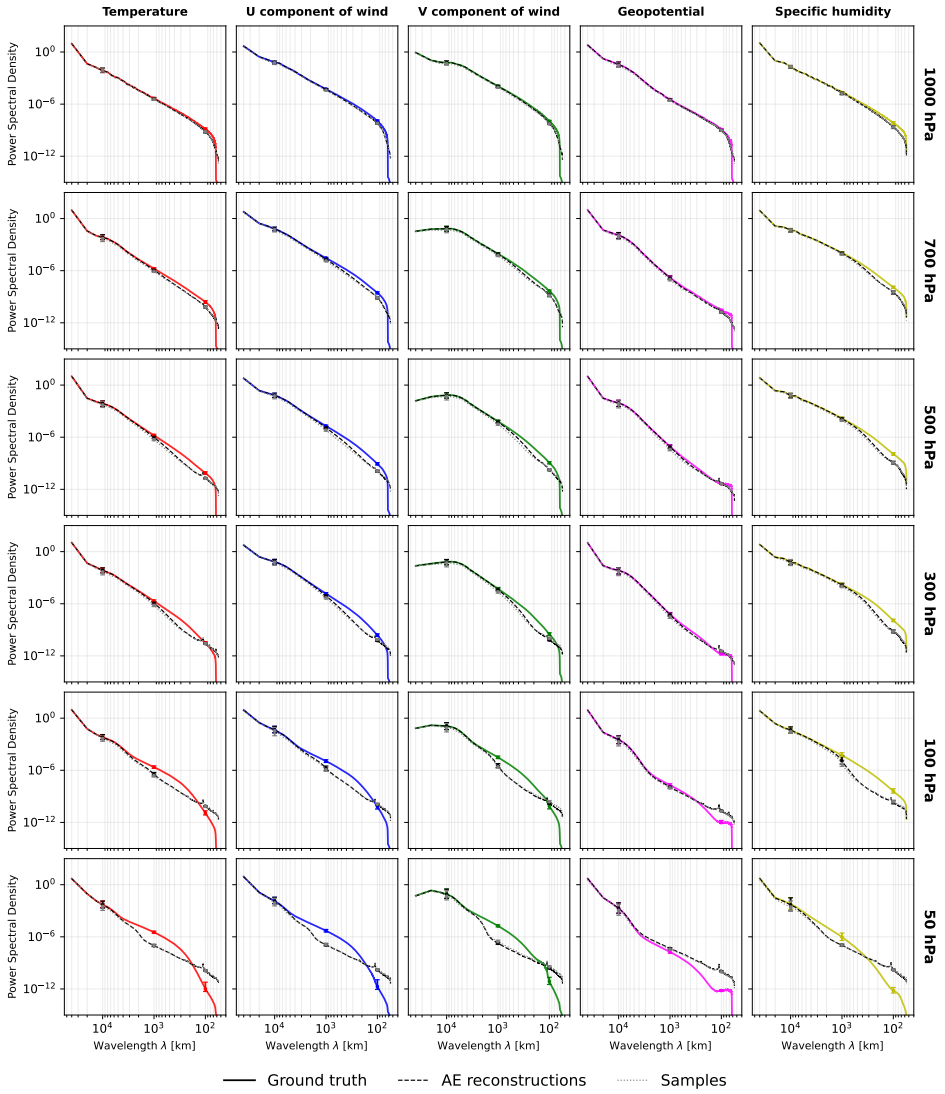


Figure 6.3. Power spectral density across wavelengths for atmospheric fields at different pressure levels between January 2021 and December 2021. Lines show median values and error bars indicate the 5th to 95th percentiles. The close alignment between the curves demonstrates that both the autoencoder and the diffusion model preserve the energy distribution across most spatial scales. Deviations begin to appear at wavelengths around 1000 km, which corresponds to roughly 40 pixels at the equator at our 0.25° resolution. These differences become more pronounced at smaller scales, suggesting that while large-scale atmospheric patterns are well-preserved, features spanning fewer than 1000 km lose energy in the compression processes. Deviations become more pronounced at lower pressure levels, as the model prioritizes surface and low-altitude fields.

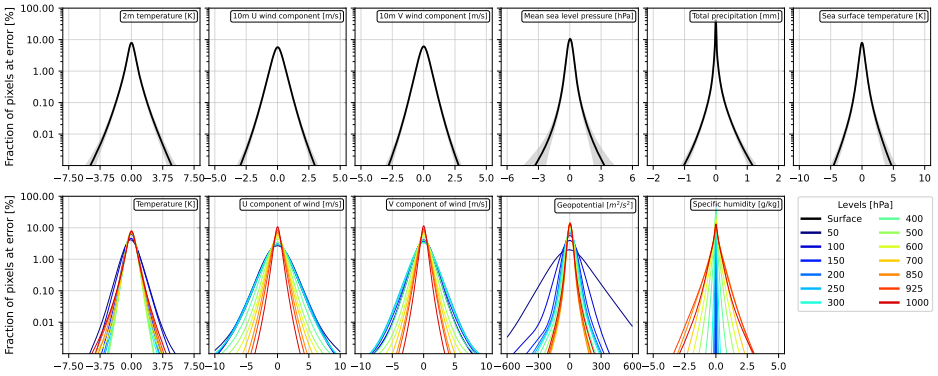


Figure 6.4. Reconstruction bias distributions for surface fields across all grid points (top) and atmospheric fields across all pressure levels (bottom). The distributions centered around zero demonstrate unbiased and accurate reconstructions. For $721 \times 1440 = 1\,038\,240$ grid points, a 0.01 % fraction on the y-axis corresponds to approximately 100 points, indicating that large biases are rare.

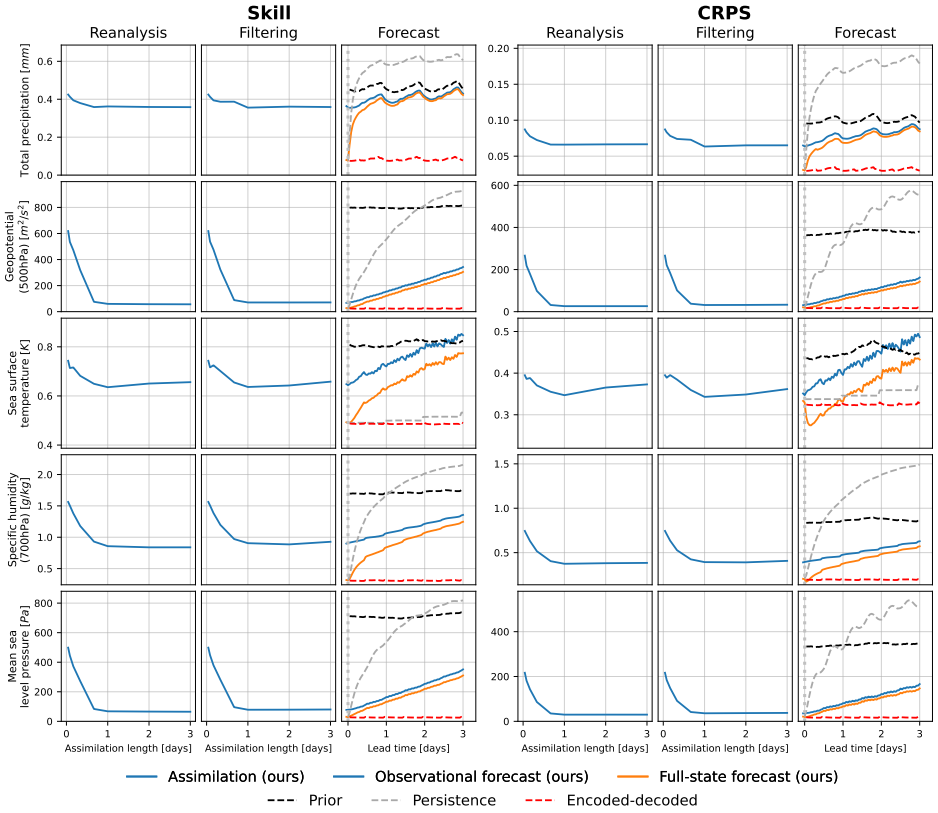


Figure 6.5. Skill and CRPS of other representative fields in assimilation tasks over January 2023. Reanalysis metrics are averaged over assimilation windows $x^{1:L}$, while filtering only considers the last assimilated state x^L. Both improve with longer assimilation windows, but eventually stagnate. Forecasts gradually lose skill over lead time. IFS [34] and GraphDOP [35] are shown for reference.

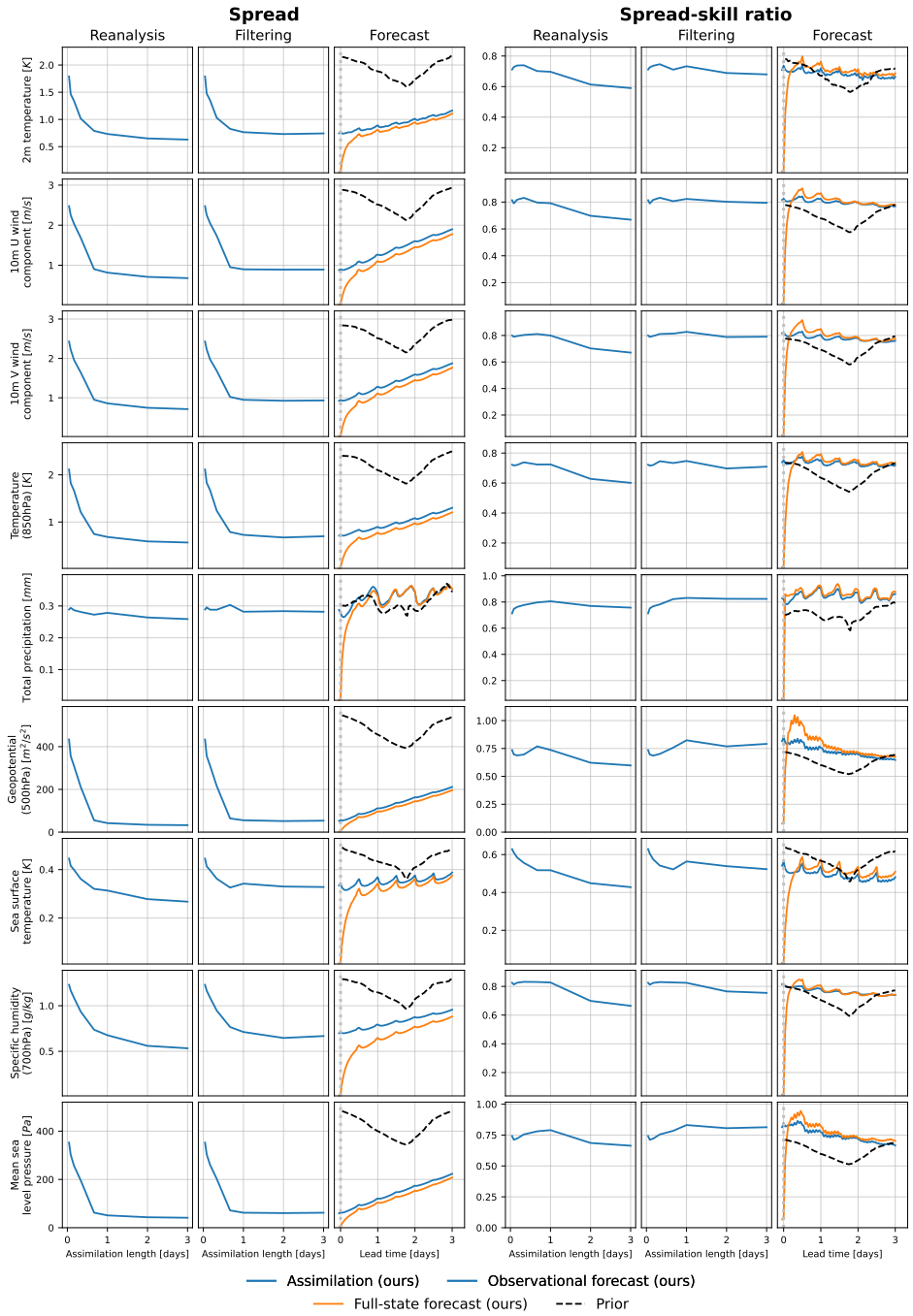


Figure 6.6. Spread and Spread-Skill ratio of representative fields in assimilation tasks over January 2023. Metrics are averaged over assimilation windows $x^{1:L}$, while filtering only considers the last assimilated state x^L. Ensemble spread (i.e. uncertainty) decreases with longer assimilation windows, while spread-skill ratios remain constant. A ratio below one indicates overconfidence.

6.D.1 Physical consistency

To further evaluate physical consistency, we examine whether our model preserves important physical relationships between variables. First, we analyze the consistency between two different estimators of altitude at given pressure levels. Using the geopotential Φ, altitude can be derived as

$$H = \frac{\Phi R_e}{g_0 R_e - \Phi}, \tag{6.13}$$

where R_e is Earth's radius and g_0 is the Earth gravitational acceleration at the surface. Alternatively, the equation below (which relies on the ideal gas law and hydrostatic equation) relates altitude to pressure and temperature as

$$\log \frac{p_0}{p_H} = \frac{M g_0}{R} \int_0^H \frac{1}{T_h} \partial h, \tag{6.14}$$

where R is the universal gas constant, M is an approximation of the atmosphere's molar mass, p_h, T_h are pressure and temperature at height h, and p_0, T_0 are the theoretical pressure and temperature at sea level. This integral can be approximated to extract H using several assumptions about the temperature profile. When comparing these two estimators, Figure 6.7 shows that our generated samples maintain the same systematic differences ΔH as seen in ground-truth data. This remarkable consistency indicates that our model successfully preserves this physical relationships between temperature, pressure, and geopotential, allowing altitude to be estimated through two independent methods with nearly identical accuracy to the original ERA5 data.

Second, we examine the geostrophic balance

$$\frac{\partial \Phi}{\partial x} = \frac{4\pi\Omega R_e}{N_x} \sin\phi \cos\phi u_g \tag{6.15}$$

$$\frac{\partial \Phi}{\partial y} = -\frac{2\pi\Omega R_e}{N_y} \sin\phi v_g \tag{6.16}$$

which is the theoretical equilibrium between pressure gradient forces and Coriolis forces that governs large-scale atmospheric motion. In the above system, ϕ is the latitude, N_x and N_y are the number of pixels along longitude and latitude, Ω is the magnitude of the Earth's angular velocity, and u_g and v_g denote the eostrophic components of the wind. In this balance, in the absence of vertical motion, friction, and isobaric curvature, wind direction should be perpendicular to geopotential gradients, with wind speed proportional to gradient magnitude. This relationship can be expressed by comparing two quantities: (1) the angle θ between wind and geopotential gradients, which should approach 90° in geostrophic conditions, and (2) the correlation between wind speed magnitude and geopotential gradient magnitude, which should approach 1 in perfect geostrophic balance. Figure 6.7 shows that our generated samples accurately reproduce both aspects of this relationship. At 500 hPa, the approximate level of non-divergence with minimal surface friction effects, both ERA5 data and our generated samples show angles concentrated around 90°. Near the surface at 1000 hPa, where additional forces become significant, both datasets show a systematic deviation in angle. Similarly, the correlation between wind speed and geopotential gradient magnitudes in our samples closely matches the patterns observed in ERA5 data, exhibiting imperfect correlation only at lower pressure levels (explained by the level-weighted training) and following the same decreasing trend as pressure increases toward the surface, where ageostrophic components become more prominent.

These results demonstrate that our latent diffusion model not only preserves the statistical properties of atmospheric fields but also maintains important physical relationships

between variables producing trajectories that are physically consistent and realistic. While these analyses confirm strong spatial consistency and physical fidelity, future work should extend our evaluation to more thoroughly quantify the temporal consistency of generated trajectories.

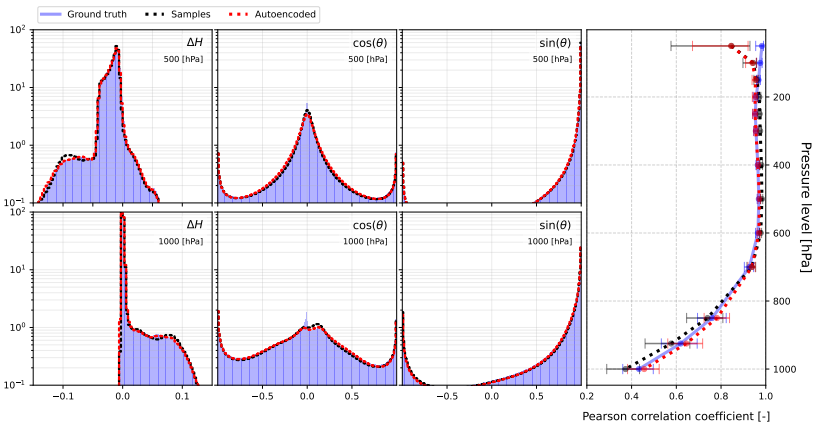


Figure 6.7. Physical consistency analysis of generated atmospheric states. **(top)** Analysis of altitude consistency at 500 hPa showing the difference ΔH between two independent altitude estimators, and geostrophic balance assessment through the cosine and sine of the angle θ between wind direction and geopotential gradients, demonstrating angles concentrated around 90°. **(bottom)** Same metrics at 1000 hPa demonstrating the presence of a significant ageostrophic component near the surface. **(right)** Correlation coefficient between wind magnitude and geopotential gradient magnitude across pressure levels, showing strong correlation at upper levels (near 1) with a consistent decrease toward the surface in both ERA5 data (blue) and generated samples (black dots), confirming Appa's ability to capture complex physical relationships.

6.D.2 Qualitative snapshots

In Figures 6.8 to 6.13, we illustrate trajectories generated through reanalysis over a window of $L = 72$ hours or through forecasting, for six representative fields. The second row of each gallery shows the observed pixels, if any. Surface fields and atmospheric fields have different observation masks (ground station network vs. satellite orbital).

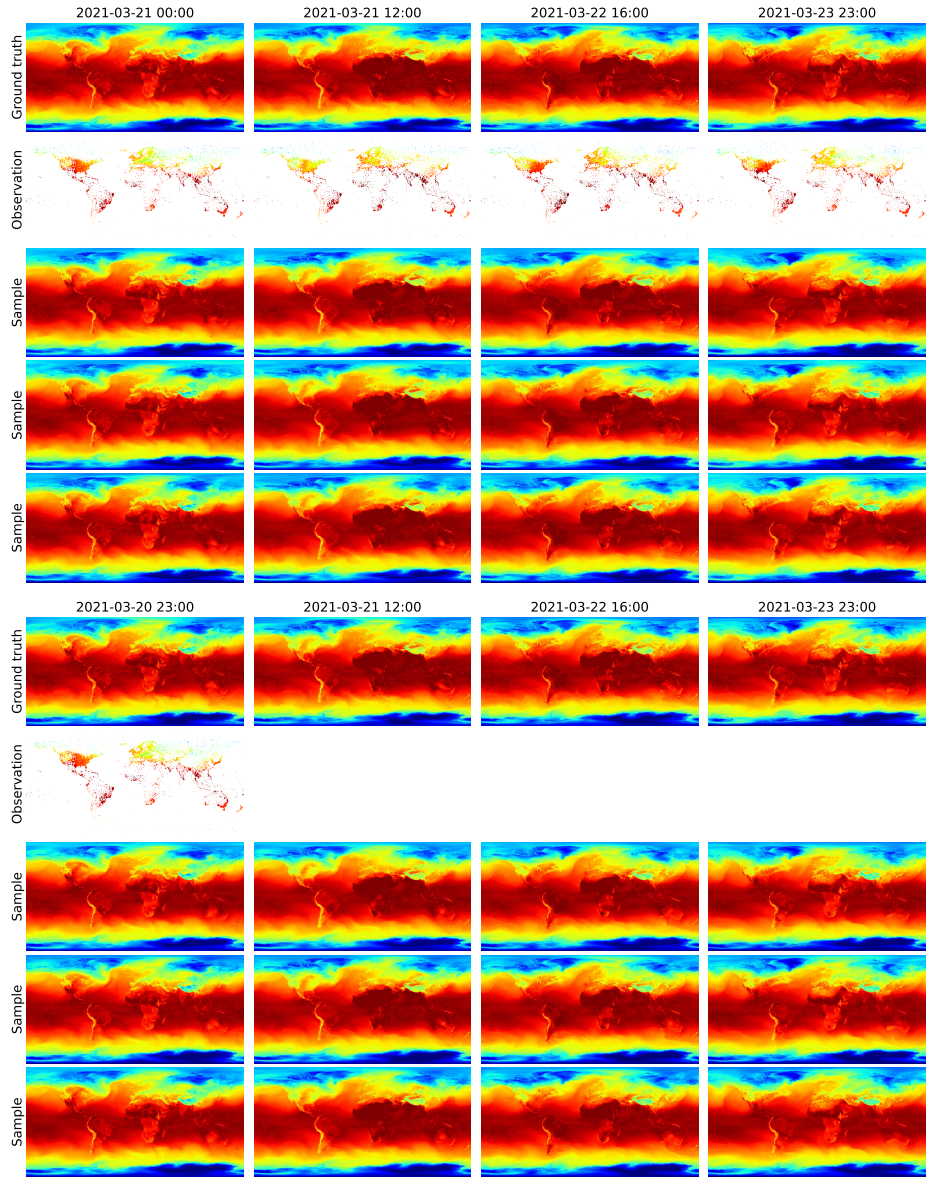


Figure 6.8. Sampled trajectories for surface temperature assimilation. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

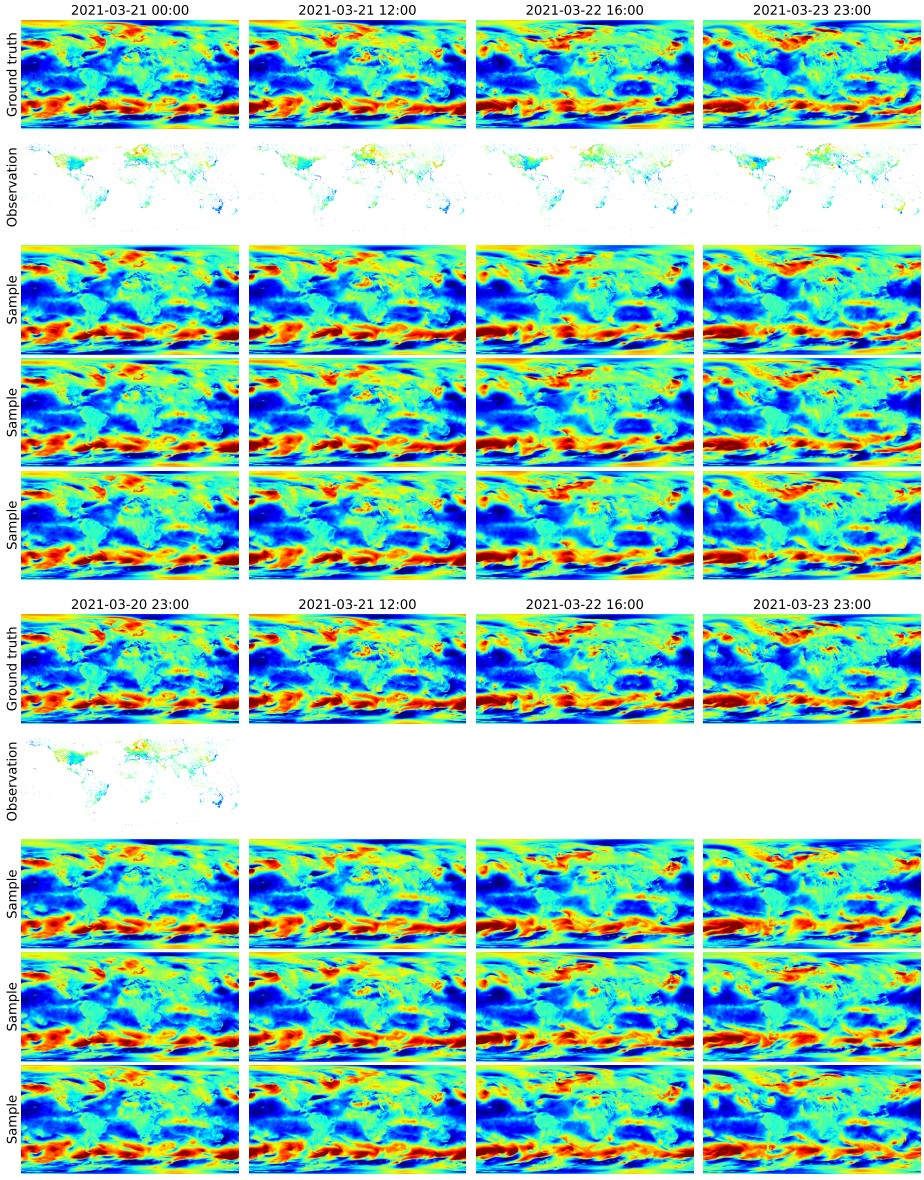


Figure 6.9. Sampled trajectories for surface zonal wind speed assimilation. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

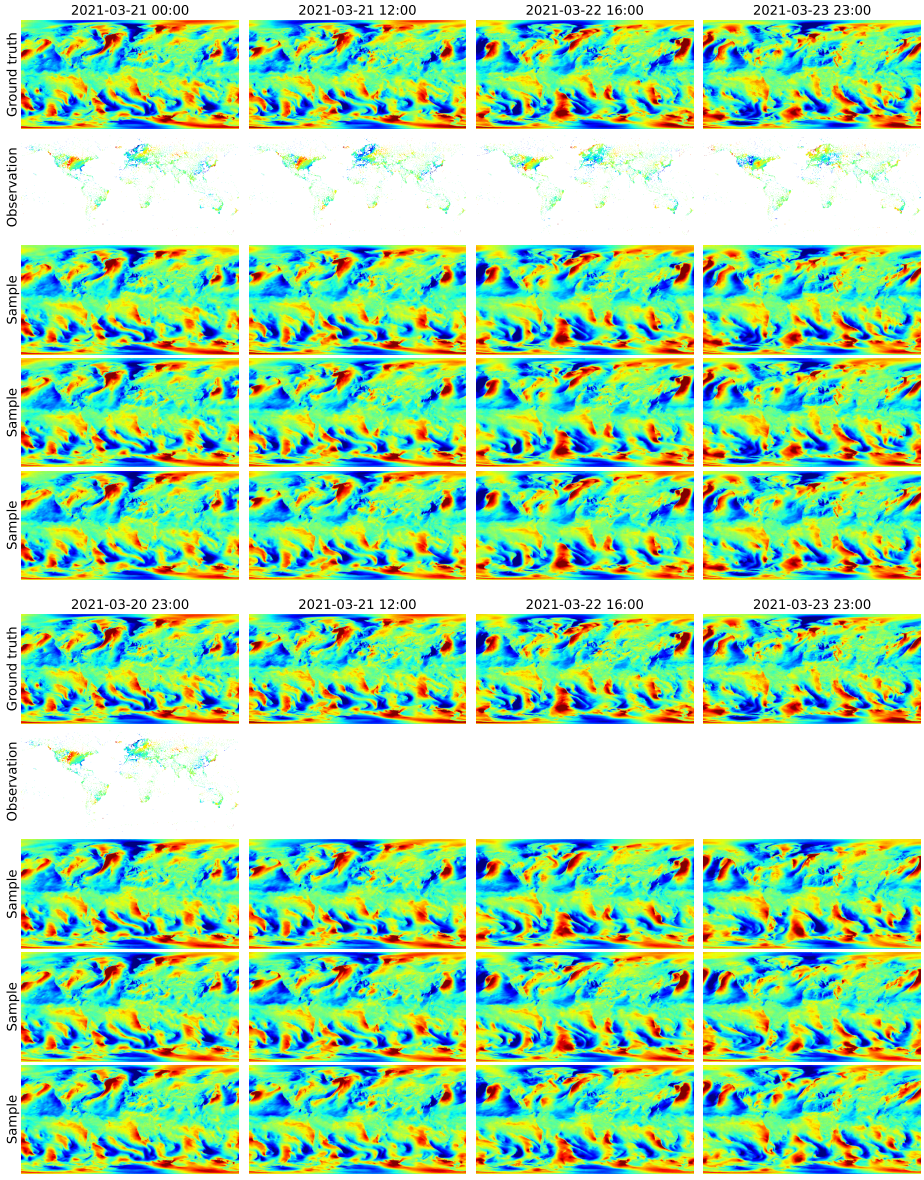


Figure 6.10. Sampled trajectories for surface meridional wind speed assimilation. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

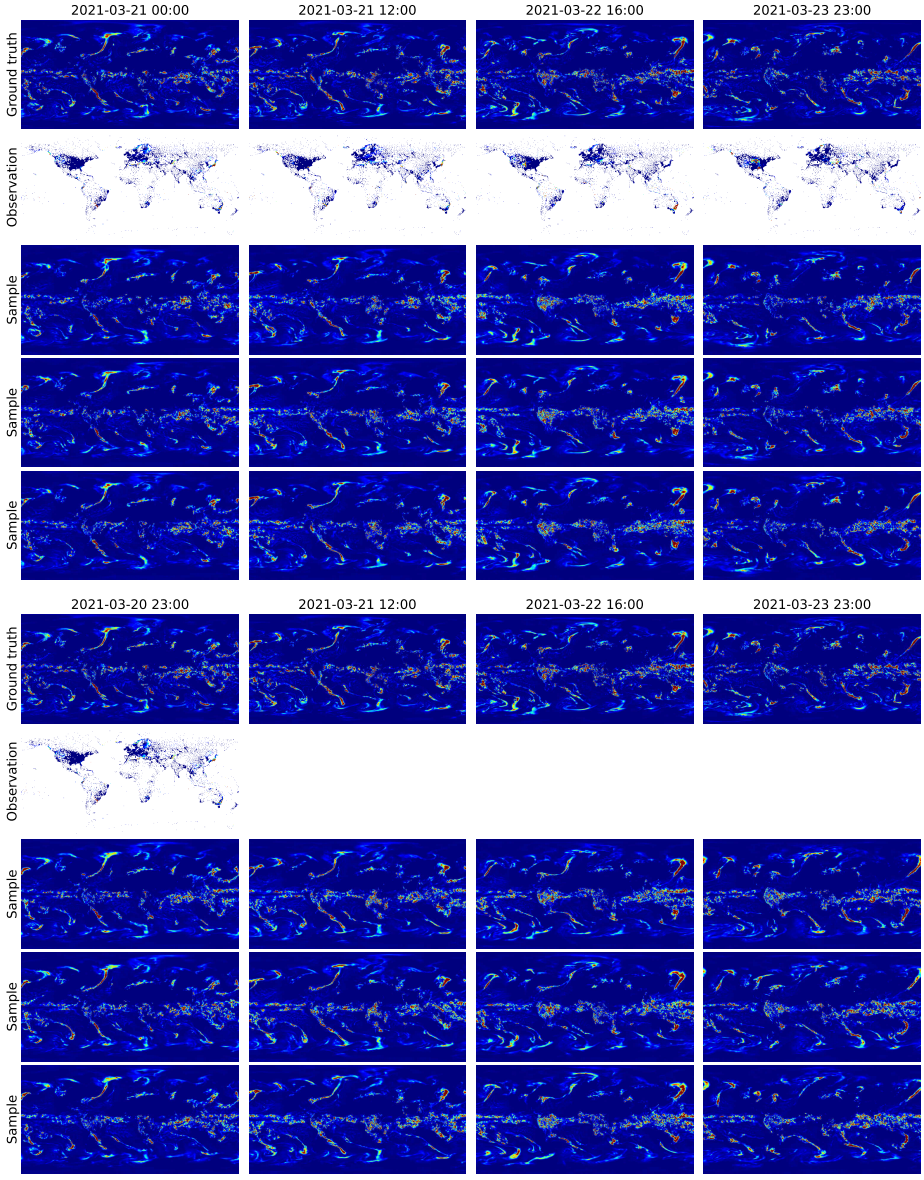


Figure 6.11. Sampled trajectories for total precipitation assimilation. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

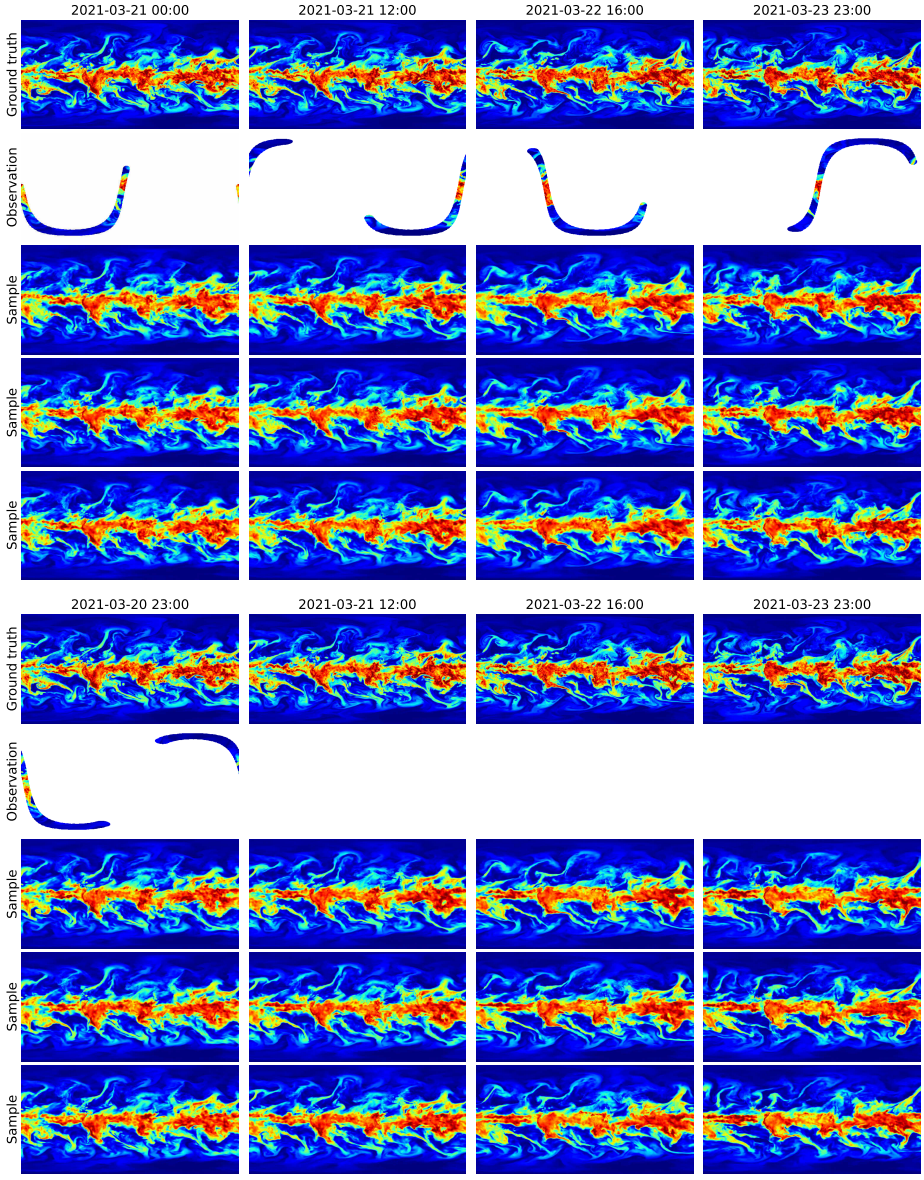


Figure 6.12. Sampled trajectories for specific humidity assimilation at 700 hPa. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

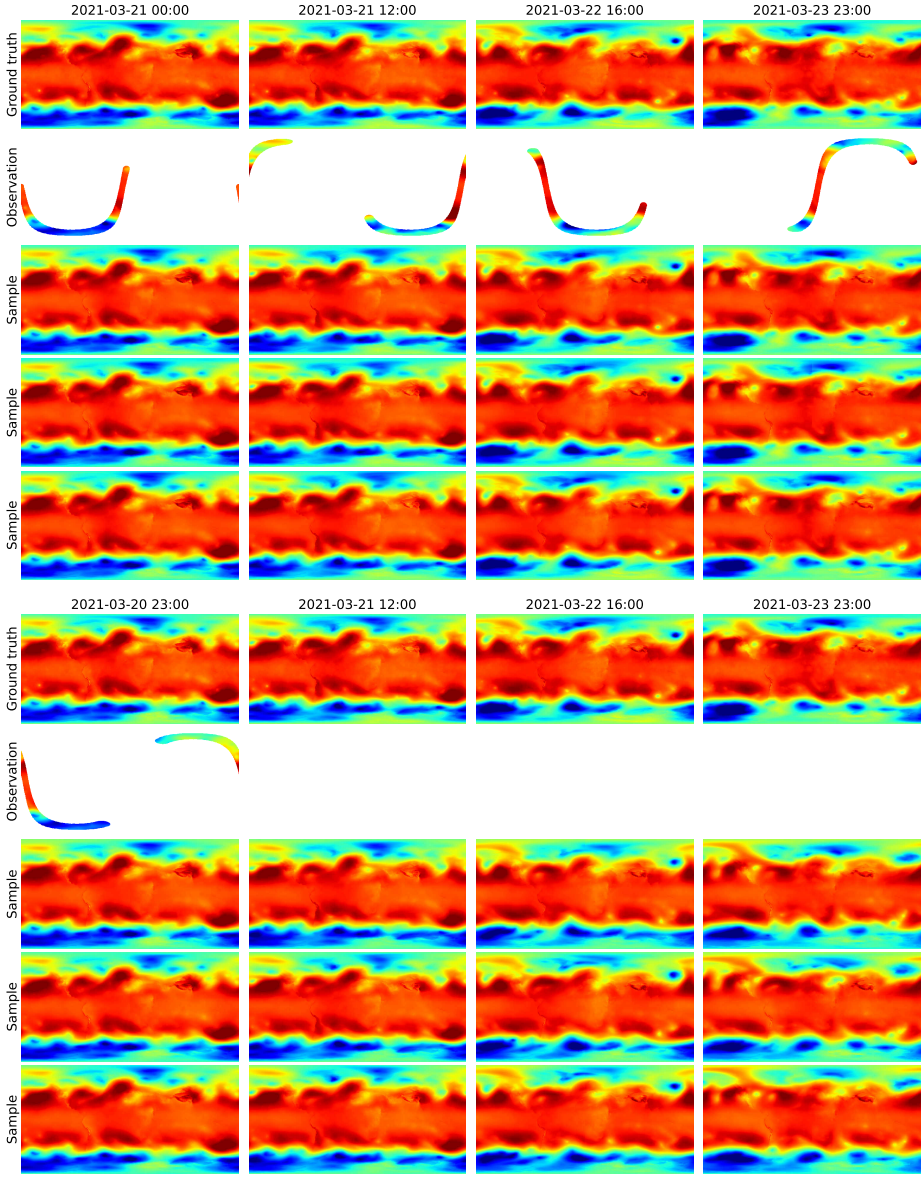


Figure 6.13. Sampled trajectories for geopotential assimilation at 850 hPa. Reanalysis (top) and observational forecasting (bottom) over a window of 72 hours.

III
Epilogue

7 Discussion

> *The view that machines cannot give rise to surprises is due, I believe, to a fallacy to which philosophers and mathematicians are particularly subject. This is the assumption that as soon as a fact is presented to a mind, all consequences of that fact spring into the mind simultaneously with it. It is a very useful assumption under many circumstances, but one too easily forgets that it is false. A natural consequence of doing so is that one then assumes that there is no virtue in the mere working out of consequences from data and general principles.*
>
> — Alan Turing (1950)

As this dissertation reaches its final chapter, we take a moment to look both backward and forward. Throughout this work, we investigate the application of generative models to large-scale inference problems in physics, focusing in particular on systems whose states evolve over time. Such dynamical systems are ubiquitous in nature, science, and engineering and thus constitute a key research area for technological and societal progress. Across the five preceding chapters, we explore multiple facets of probabilistic modeling and dynamical systems, including state estimation, forecasting, reduced-order modeling, and learning from corrupted observations. We now pause to reflect on what has been accomplished, what has been learned, and what lies ahead.

7.1 Impact

The cornerstone of our work, introduced in Chapter 2 and extended in Chapters 3 and 6, is a novel framework that combines diffusion models [2, 3] and training-free guidance methods [4–6] to estimate the state of dynamical systems from a series of observations over time; a task known as data assimilation (DA) [7–11]. Unlike classical DA methods, our method does not require simulating or differentiating through a hand-crafted physical model at inference, and is able to assimilate any observation. While previous studies combine machine learning methods and DA [12–20], we are the first to consider generative models, especially diffusion models, in this context. Following the publication of SDA [21], many studies applied generative models to DA [22–30]. We are confident that this trend will continue as generative modeling offers a principled way to handle uncertainty, unlike artificially perturbing observations to obtain an ensemble of solutions [31–34].

While SDA establishes diffusion models as a promising alternative to classical DA methods, the data it relies on during training – fully observed state sequences – is rarely accessible. In practice, only incomplete and noisy observations of the states are available, especially in earth and space sciences where the systems of interest can only be probed superficially. This limitation motivates our second contribution, presented in Chapter 4, an adaptation of the expectation-maximization algorithm [35–39] tailored to diffusion models. With our method, named diffusion-based expectation-maximization (DiEM), it becomes possible to train or tune diffusion models when only corrupted data is available. Remarkably, the learned diffusion model can later be used as prior in Bayesian inference problems,

such as gravitational lensing inversion [40–42], accelerated MRI [43–47], or ... data assimilation [7–11]. Although this research question has been approached from different directions [48–51], learning from corrupted data remains an underexplored research area, with virtually endless applications and unparalleled potential for impact.

Orthogonal to data availability, computational cost is another limitation of SDA, which becomes prohibitive for long trajectories and high-dimensional states, although it remains cheaper than 4D-Var [8, 52]. Our last contribution, unifying the findings of Chapters 5 and 6, is to demonstrate that modeling dynamical systems in a compressed latent representation is a sensible approach to reduce training and inference cost. Latent models are not only more efficient, but can also improve the accuracy of emulation and assimilation compared to physical-space models. In the context of numerical weather prediction, this efficiency could enable assimilation of a larger volume of observations or include physical processes that are currently omitted due to their computational burden, such as the interaction between the atmosphere and oceans.

7.2 Limitations & Perspectives

While the results presented in this dissertation are promising, several methodological and experimental limitations remain. Each of these limitations represents an exciting opportunity for future research, and addressing them will be essential to accelerate the adoption of generative modeling and, more broadly, deep learning by domain scientists. We organize our discussion around three axes: long-term dependencies, posterior validation, and learning from observations.

The frontier of long-term assimilation

A first limitation concerns SDA's ability to capture long-term dependencies. The pseudo-blanket approximation (2.13), although accurate for high signal-to-noise regimes ($\sigma_t \to 0$), introduces errors as the noise magnitude increases. In retrospect, evaluating this approximation on simplified problems would have provided valuable insight. Later studies [53] revealed that this approximation hinders the propagation of local information beyond the temporal receptive field of the model. This limitation becomes problematic for systems governed by slowly evolving dynamics, where distant past observations strongly constrain present trajectories.

The perturbations which the motions of planets suffer from the influence of other planets, are so small and so slow that they only become sensible after a long interval of time.

— C. F. Gauss (1809)

In forecasting tasks, where observations beyond the present-time cutoff are unavailable, Shysheya et al. [53] demonstrate that an autoregressive generation strategy effectively extends the range of information propagation. Their findings are consistent with our own forecasting experiments in Chapter 6. In the case of filtering, this approach is also applicable as past observations can be assimilated autoregressively. Conversely, autoregression is inapplicable for reanalysis, where both past and future observations must be assimilated simultaneously.

Within the SDA framework, we foresee several approaches to address this persistent limitation. First, training models with wider temporal windows can improve assimilation quality, especially when observations are dense [53]. However, this strategy quickly becomes computationally expensive during training and inference. A related approach would be to coarsen the temporal resolution of the model by increasing the time stride between consecutive window elements. Such model would have to understand long-term dependencies to denoise effectively, while reducing the computational cost compared to the previous proposition. Unfortunately, with this approach, it becomes difficult to assimilate high-cadence observations, whose corresponding states are not present in a coarse trajectory. A model with heterogeneous windows, containing a mixture of dense-local and sparse-global elements, may strike the right balance between fine and

coarse temporal resolutions, offering an efficient way to propagate local information globally. In practice, there is a risk that such model learns to ignore the global elements when denoising the local ones, defeating the original purpose.

Beyond architectural modifications, the path to accurate long-term modeling could also stem from algorithmic improvements. A recent line of work [55–57] demonstrates that using time-correlated noise, instead of white noise, can transform a pre-trained image diffusion model into a naive video model. Adapting this strategy to the SDA framework could mitigate the issue of long-term decorrelation, but is unlikely to solve it. From another perspective, Du et al. [58] describe several methods to compose the outputs of multiple diffusion models during generation. These methods are different from the one presented in Chapter 2 and could lead to better information propagation. Finally, efficient global models could replace local models when the latter become too imprecise. Notably, in low signal-to-noise regimes, analytical denoisers derived from the data's mean and covariance [59] can outperform local neural network-based denoisers, as they attend to the entire trajectory rather than finite windows. Implementing such an analytical denoiser is technically straightforward and should therefore be prioritized for investigation.

Together, these directions highlight a broader open question [60–64] in deep learning: how to encourage long-range understanding in neural networks?

Posterior approximations can be unfaithful

In Bayesian inference, the posterior distribution encapsulates all the knowledge we possess about the problem at hand. If a posterior approximation is too inaccurate, it can lead to misleading conclusions that can propagate through downstream analysis and eventually influence policy or operational decisions. Therefore, posterior validation is not only a matter of scientific rigor, but a necessity to establish trust [65] in deep probabilistic models. Unfortunately, this aspect is often overlooked within the literature, and this dissertation is no exception.

The main reason for this frequent shortcoming is the overall lack of simultaneously principled and scalable diagnostics. Widespread metrics in the image generation literature, such as PSNR, SSIM [66], and LPIPS [67] are cheap to evaluate but poorly aligned with posterior faithfulness: PSNR and SSIM favor smooth reconstructions, whereas LPIPS emphasizes perceptual similarity. FID [68] is sound but applies only to unconditional generation, as the true conditional/posterior distribution is typically unavailable. Conversely, common posterior diagnostics in the Bayesian inference literature, such as expected coverage probability (ECP) of highest posterior density regions [65, 69–71], are rigorous but expensive in high dimensions, rely on probability density estimates, and fail to detect uninformative posteriors [65, 72]. Recent sample-based alternatives such as TARP [72] or PQMass [73] appear to be more robust, but remain computationally demanding. In meteorology, the continuous ranked probability score (CRPS) [74, 75] is both principled and scalable, but only probes one-dimensional marginals. Multivariate scoring rules exist [75], but their discriminative power is notoriously limited [76]. Finally, most metrics are meaningful only in expectation over multiple observations and, thus, not appropriate for case-by-case decision making. The local classifier two-sample test (ℓ-C2ST) proposed by Linhart et al. [77] enables localized posterior validation for individual observations, but relies on an auxiliary classifier trained with held-out calibration data and remains untested in high-dimensional regimes.

In other words, developing evaluation diagnostics that are accessible, principled, scalable, and useful remains an open challenge in making deep probabilistic models scientifically trustworthy and actionable.

Mining gold from observations

Beyond methodological and experimental shortcomings, an aspect missing from this dissertation is the integration of the DiEM algorithm, presented in Chapter 4, within the SDA framework, introduced in Chapter 2. Combining the two could, in principle, enable learning global atmospheric models directly from observational data, thereby circumventing the need for synthetic data and hand-crafted models that typically misrepresent sub-resolution phenomena. This approach recalls the iterative strategy proposed by Brajard et al. [13, 14]; an implicit instance of the EM algorithm, where data assimilation acts as the expectation step, while training an emulator of the dynamics corresponds to the maximization step. In our work, we either rely on reanalysis data from ERA5 [78] or synthetic data from a simulator to train models. In both cases, our models inherit the biases of the underlying physical models, and become misspecified with respect to the true dynamics. Detecting and mitigating model misspecification in Bayesian inference is an active area of research [79–94] and refining misspecified diffusion models with DiEM is a promising approach, adopted by Barco et al. [95].

Tant que l'Algèbre et la Géométrie ont été séparées, leurs progrès ont été lents et leurs usages bornés; mais lorsque ces deux sciences se sont réunies, elles se sont prêté des forces mutuelles et ont marché ensemble d'un pas rapide vers la perfection.

— J.-L. Lagrange (1795)

Another long-standing obstacle in data assimilation, inherited by SDA, is to exploit observations for which no explicit and differentiable model of the measurement process exists, such as satellite data. Previous studies [19, 96, 97] have tackled this issue by training end-to-end models to predict one modality of observations from another modality. These purely data-driven approaches automatically compensate for unknown phenomena or biases in the data, at the expense of flexibility: incorporating a new type of observation typically requires retraining. This stands in contrast to the modular, plug-and-play nature of training-free guidance methods [4–6], central to the SDA framework.

An intermediate solution, compatible with SDA, would be to train differentiable surrogate models for the unknown measurement processes (or their inverse). Such surrogates could then be naturally integrated into the perturbed likelihood approximations of Chapters 2 or 6. However, training them requires paired state-observation data, which is rarely accessible in earth sciences. A possible remedy is to integrate this training step within the EM algorithm, using samples from the expectation step to refine surrogate measurement models, in the spirit of Gibbs sampling [98, 99]. This *blind* EM scheme could, in principle, jointly learn a prior model for the dynamics and surrogate models for the unknown measurement processes.

Whether through measurement surrogates, end-to-end models, or EM-like methods, considerable investments remain to be made for integrating a larger fraction of observational data within NWP pipelines.

7.3 Conclusion

Ultimately, this thesis argues for a more probabilistic view of dynamical systems and scientific modeling; one that acknowledges uncertainty in observations, states, and models. This objective invites to move beyond the pitfalls of hand-crafted models and embrace data-driven methods, that can learn automatically without inheriting our biases. Our work takes a step forward in this direction by showing that generative models and, in particular, diffusion models can replicate, discover, and abstract the dynamics of our universe, solely from data. Many may oppose our position as they see machines as mere number jugglers, incapable of producing new knowledge. But, as Turing [1] reminds us, there is value in the systematic derivation of consequences from data. Machines do so tirelessly, and the fruits of their labor surprise us with great frequency.

References

[1] Alan Turing. "Computing Machinery and Intelligence". In *Mind* LIX.236 (1950).

[2] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020.

[4] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In *International Conference on Learning Representations*. 2023.

[5] Jiaming Song et al. "Pseudoinverse-Guided Diffusion Models for Inverse Problems". In *International Conference on Learning Representations*. 2023.

[6] Benjamin Boys et al. "Tweedie Moment Projected Diffusions For Inverse Problems". 2023.

[7] François-Xavier Le Dimet and Olivier Talagrand. "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects". In *Tellus A* 38A.2 (1986).

[8] Yannick Trémolet. "Accounting for an imperfect model in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 132.621 (2006).

[9] Thomas M. Hamill. "Ensemble-based atmospheric data assimilation". In *Predictability of Weather and Climate*. 2006.

[10] Geir Evensen. "Data Assimilation: The Ensemble Kalman Filter". Springer, 2009.

[11] Alberto Carrassi et al. "Data assimilation in the geosciences: An overview of methods, issues, and perspectives". In *WIREs Climate Change* 9 (2018).

[12] Julian Mack et al. "Attention-based Convolutional Autoencoders for 3D-Variational Data Assimilation". In *Computer Methods in Applied Mechanics and Engineering* 372 (2020).

[13] Julien Brajard et al. "Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model". In *Journal of Computational Science* 44 (2020).

[14] Julien Brajard et al. "Combining data assimilation and machine learning to infer unresolved scale parametrization". In *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (2021).

[15] Rossella Arcucci et al. "Deep Data Assimilation: Integrating Deep Learning with Data Assimilation". In *Applied Sciences* 11.3 (2021).

[16] R. Fablet et al. "Learning Variational Data Assimilation Models and Solvers". In *Journal of Advances in Modeling Earth Systems* 13.10 (2021).

[17] Thomas Frerix et al. "Variational Data Assimilation with a Learned Inverse Observation Operator". In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[18] Yu-Hong Yeung, David A. Barajas-Solano, and Alexandre M. Tartakovsky. "Physics-Informed Machine Learning Method for Large-Scale Data Assimilation Problems". In *Water Resources Research* 58.5 (2022).

[19] Marcin Andrychowicz et al. "Deep Learning for Day Forecasts from Sparse Observations". 2023.

[20] Sibo Cheng et al. “Machine Learning With Data Assimilation and Uncertainty Quantification for Dynamical Systems: A Review”. In *IEEE/CAA Journal of Automatica Sinica* 10.6 (2023).

[21] François Rozet and Gilles Louppe. “Score-based Data Assimilation”. In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[22] Langwen Huang et al. “DiffDA: a Diffusion model for weather-scale Data Assimilation”. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[23] Ehsan Foroumandi and Hamid Moradkhani. “Harnessing Generative Deep Learning for Enhanced Ensemble Data Assimilation”. In *Water Resources Research* 61.7 (2025).

[24] Scott A. Martin, Georgy E. Manucharyan, and Patrice Klein. “Generative Data Assimilation for Surface Ocean State Estimation From Multi-Modal Satellite Observations”. In *Journal of Advances in Modeling Earth Systems* 17.8 (2025).

[25] Peter Manshausen et al. “Generative Data Assimilation of Sparse Weather Station Observations at Kilometer Scales”. 2025.

[26] Jonathan Schmidt et al. “A Generative Framework for Probabilistic, Spatiotemporally Coherent Downscaling of Climate Simulation”. In *npj Climate and Atmospheric Science* 8.1 (2025).

[27] Zheqi Shen. “Conditional Denoising Score Matching for Sequential Data Assimilation”. In *Ocean-Land-Atmosphere Research* 4 (2025).

[28] Jing-An Sun et al. “Align-DA: Align Score-based Atmospheric Data Assimilation with Multiple Preferences”. 2025.

[29] Dibyajyoti Chakraborty et al. “Multimodal Atmospheric Super-Resolution With Deep Generative Models”. 2025.

[30] Xiaoze Xu et al. “FuXi-DA: a generalized deep learning data assimilation framework for assimilating satellite observations”. In *npj Climate and Atmospheric Science* 8.1 (2025).

[31] Chengsi Liu and Qingnong Xiao. “An Ensemble-Based Four-Dimensional Variational Data Assimilation Scheme. Part III: Antarctic Applications with Advanced Research WRF Using Real Data”. In *Monthly Weather Review* 141.8 (2013).

[32] I. Hoteit et al. “Mitigating Observation Perturbation Sampling Errors in the Stochastic EnKF”. In *Monthly Weather Review* 143.7 (2015).

[33] Thomas Auligné et al. “Ensemble–Variational Integrated Localized Data Assimilation”. In *Monthly Weather Review* 144.10 (2016).

[34] R. N. Bannister. “A review of operational methods of variational and ensemble-variational data assimilation”. In *Quarterly Journal of the Royal Meteorological Society* 143.703 (2017).

[35] H. O. Hartley. “Maximum Likelihood Estimation from Incomplete Data”. In *Biometrics* (1958).

[36] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In *Journal of the Royal Statistical Society* (1977).

[37] C. F. Jeff Wu. “On the Convergence Properties of the EM Algorithm”. In *The Annals of Statistics* (1983).

[38] Geoffrey J McLachlan and Thriyambakam Krishnan. "The EM algorithm and extensions". John Wiley & Sons, 2007.

[39] Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. "Statistical guarantees for the EM algorithm: From population to sample-based analysis". In *The Annals of Statistics* (2017).

[40] S. J. Warren and S. Dye. "Semilinear Gravitational Lens Inversion". In *The Astrophysical Journal* (2003).

[41] Warren R. Morningstar et al. "Data-driven Reconstruction of Gravitationally Lensed Galaxies Using Recurrent Inference Machines". In *The Astrophysical Journal* (2019).

[42] Siddharth Mishra-Sharma and Ge Yang. "Strong Lensing Source Reconstruction Using Continuous Neural Fields". 2022.

[43] Shanshan Wang et al. "Accelerating magnetic resonance imaging via deep learning". In *International Symposium on Biomedical Imaging*. 2016.

[44] Kerstin Hammernik et al. "Learning a variational network for reconstruction of accelerated MRI data". In *Magnetic Resonance in Medicine* (2018).

[45] Yoseo Han, Leonard Sunwoo, and Jong Chul Ye. "k-Space Deep Learning for Accelerated MRI". In *Transactions on Medical Imaging* (2020).

[46] Jure Zbontar et al. "fastMRI: An Open Dataset and Benchmarks for Accelerated MRI". 2018.

[47] Florian Knoll et al. "fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning". In *Radiology: Artificial Intelligence* (2020).

[48] Asad Aali et al. "Solving Inverse Problems with Score-Based Generative Priors learned from Noisy Data". In *2023 57th Asilomar Conference on Signals, Systems, and Computers*. 2023.

[49] Giannis Daras et al. "Ambient Diffusion: Learning Clean Distributions from Corrupted Data". In *Advances in Neural Information Processing Systems*. Vol. 36. 2023.

[50] Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. "Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data". In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[51] Bahjat Kawar et al. "GSURE-Based Diffusion Model Training with Corrupted Data". In *Transactions on Machine Learning Research* (2024).

[52] Yannick Trémolet. "Model-error estimation in 4D-Var". In *Quarterly Journal of the Royal Meteorological Society* 133.626 (2007).

[53] Aliaksandra Shysheya et al. "On conditional diffusion models for PDE simulations". In *Advances in Neural Information Processing Systems*. Vol. 37. 2024.

[54] Carl Friedrich Gauss. "Theoria motus corporum coelestium in sectionibus conicis solem ambientium". 1809.

[55] Songwei Ge et al. "Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models". In *IEEE/CVF International Conference on Computer Vision*. 2023.

[56] Pascal Chang et al. "How I Warped Your Noise: a Temporally-Correlated Noise Prior for Diffusion Models". In *International Conference on Learning Representations*. 2024.

[57] Giannis Daras et al. “Warped Diffusion: Solving Video Inverse Problems with Image Diffusion Models”. In *Advances in Neural Information Processing Systems* 37 (2024).

[58] Yilun Du et al. “Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC”. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[59] Simo Särkkä. “Bayesian Filtering and Smoothing”. Cambridge University Press, 2013.

[60] Iz Beltagy, Matthew E. Peters, and Arman Cohan. “Longformer: The Long-Document Transformer”. 2020.

[61] Yi Tay et al. “Long Range Arena : A Benchmark for Efficient Transformers”. In *International Conference on Learning Representations*. 2021.

[62] Albert Gu, Karan Goel, and Christopher Re. “Efficiently Modeling Long Sequences with Structured State Spaces”. In *International Conference on Learning Representations*. 2022.

[63] Vijay Prakash Dwivedi et al. “Long Range Graph Benchmark”. In *Advances in Neural Information Processing Systems* 35 (2022).

[64] Jianlin Su et al. “RoFormer: Enhanced transformer with Rotary Position Embedding”. In *Neurocomputing* 568 (2024).

[65] Joeri Hermans et al. “A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful”. In *Transactions on Machine Learning Research* (2022).

[66] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In *Transactions on Image Processing* (2004).

[67] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.

[68] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[69] François Rozet. “Arbitrary Marginal Neural Ratio Estimation for Likelihood-free Inference”. PhD thesis. Université de Liège, 2021.

[70] Benjamin K Miller et al. “Truncated Marginal Neural Ratio Estimation”. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021.

[71] Michael Deistler, Pedro J. Goncalves, and Jakob H. Macke. “Truncated proposals for scalable and hassle-free simulation-based inference”. In *Advances in Neural Information Processing Systems* 35 (2022).

[72] Pablo Lemos et al. “Sampling-Based Accuracy Testing of Posterior Estimators for General Inference”. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[73] Pablo Lemos et al. “PQMass: Probabilistic Assessment of the Quality of Generative Models using Probability Mass Estimation”. In *International Conference on Learning Representations*. 2025.

[74] Hans Hersbach. “Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems”. In *Weather and Forecasting* 15.5 (2000).

[75] Tilmann Gneiting and Adrian E Raftery. "Strictly Proper Scoring Rules, Prediction, and Estimation". In *Journal of the American Statistical Association* 102.477 (2007).

[76] Pierre Pinson and Julija Tastu. "Discrimination ability of the Energy score". Report. Technical University of Denmark, 2013.

[77] Julia Linhart, Alexandre Gramfort, and Pedro Rodrigues. "L-C2ST: Local Diagnostics for Posterior Approximations in Simulation-Based Inference". In *Advances in Neural Information Processing Systems* 36 (2023).

[78] Hans Hersbach et al. "The ERA5 global reanalysis". In *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020).

[79] Badr-Eddine Cherief-Abdellatif and Pierre Alquier. "MMD-Bayes: Robust Bayesian Estimation via Maximum Mean Discrepancy". In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*. PMLR, 2020.

[80] Antoine Wehenkel et al. "Robust Hybrid Learning With Expert Augmentation". In *Transactions on Machine Learning Research* (2022).

[81] Charita Dellaporta et al. "Robust Bayesian Inference for Simulator-based Models via the MMD Posterior Bootstrap". In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

[82] Patrick Cannon, Daniel Ward, and Sebastian M. Schmon. "Investigating the Impact of Model Misspecification in Neural Simulation-based Inference". 2022.

[83] Daniel Ward et al. "Robust Neural Posterior Estimation and Statistical Model Criticism". In *Advances in Neural Information Processing Systems* 35 (2022).

[84] Daolang Huang et al. "Learning Robust Statistics for Simulation-based Inference under Model Misspecification". In *Advances in Neural Information Processing Systems* 36 (2023).

[85] David T. Frazier et al. "Reliable Bayesian Inference in Misspecified Models". 2023.

[86] Richard Gao, Michael Deistler, and Jakob H. Macke. "Generalized Bayesian Inference for Scientific Simulators via Amortized Cost Estimation". In *Advances in Neural Information Processing Systems* 36 (2023).

[87] David J. Nott, Christopher Drovandi, and David T. Frazier. "Bayesian Inference for Misspecified Generative Models". In *Annual Review of Statistics and Its Application* 11.Volume 11, 2024 (2024).

[88] Marvin Schmitt et al. "Detecting Model Misspecification in Amortized Bayesian Inference with Neural Networks". In *Pattern Recognition*. Springer Nature Switzerland, 2024.

[89] Ryan P. Kelly et al. "Misspecification-robust Sequential Neural Likelihood for Simulation-based Inference". In *Transactions on Machine Learning Research* (2024).

[90] Noemi Anau Montel, James Alvey, and Christoph Weniger. "Tests for model misspecification in simulation-based inference: From local distortions to global model checks". In *Physical Review D* 111.8 (2025).

[91] Sébastien Pierre et al. "Mitigating Model Misspecification in Simulation-Based Inference for Galaxy Clustering". 2025.

[92] Antoine Wehenkel et al. "Addressing Misspecification in Simulation-based Inference through Data-driven Calibration". In *International Conference on Machine Learning*. 2025.

[93] Ortal Senouf et al. "Inductive Domain Transfer In Misspecified Simulation-Based Inference". 2025.

[94] Jens Behrmann et al. “Inferring Optical Tissue Properties from Photoplethysmography using Hybrid Amortized Inference”. 2025.

[95] Gabriel Missael Barco et al. “Tackling the Problem of Distributional Shifts: Correcting Misspecified, High-dimensional Data-driven Priors for Inverse Problems”. In *The Astrophysical Journal* 980.1 (2025).

[96] Mihai Alexe et al. “GraphDOP: Towards skilful data-driven medium-range weather forecasts learnt and initialised directly from observations”. 2024.

[97] Anna Allen et al. “End-to-end data-driven weather prediction”. In *Nature* 641.8065 (2025).

[98] Stuart Geman and Donald Geman. “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (1984).

[99] David Heurtel-Depeiges et al. “Listening to the noise: Blind Denoising with Gibbs Diffusion”. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.