Reducing Memorisation in Generative Models via Riemannian Bayesian Inference

Johanna Marie Gegenfurtner*
Technical University of Denmark
johge@dtu.dk

Albert Kjøller Jacobsen*
Technical University of Denmark
akjja@dtu.dk

Georgios Arvanitidis
Technical University of Denmark
gear@dtu.dk

Abstract

How to balance memorisation and generalisation in generative models remains an open task. To investigate this, we employ Bayesian methods, which have recently been proposed to predict the uncertainty of generated samples. In our work, we employ the Riemannian Laplace approximation, from which we can sample generative models that resemble the trained one. Our geometry-aware approach yields improved results compared to the Euclidean counterpart.

1 Introduction

The success of modern generative models has raised questions about their capacity to merely memorise data or generate beyond the latter. While it is essential that a generative model captures the data distribution, it is critical in several applications to avoid overfitting to specific training examples. Our work focusses on diffusion models, for which the problem of memorisation has been extensively discussed in recent work, raising concerns about privacy and copyright infringement [2, 4, 6, 7, 14, 18].

In this paper, we raise the question:

Can we reduce memorisation in modern generative models through uncertainty on their parameters?

To mitigate memorisation in modern generative models, we adopt a Bayesian treatment of the model parameters, using the Laplace approximation (LA) for defining a conceptually simple approximate posterior distribution. The Laplace approximation is defined as a Gaussian centred at the maximum a posteriori (MAP) estimate, however, it is often an overly crude approximation of the true posterior [1]. To address this issue, we make use of the Riemannian Laplace approximation [1, 21] that leverages the geometric

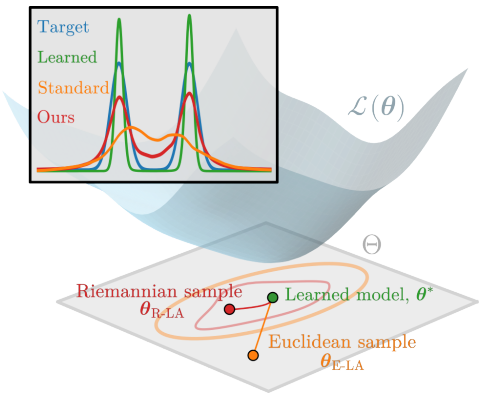

Figure 1: A generative model can learn to memorise the data seen during training. We propose reducing memorisation of a trained generative model (●) by adding noise to the learnt model parameters $\boldsymbol{\theta}^*$ through sampling from an approximate posterior distribution $\boldsymbol{\theta} \sim q(\boldsymbol{\theta})$. While a Euclidean approximate posterior (—) reduces memorisation, accounting for the geometry (—) *reduces memorisation without breaking the fit.*

*Equal contribution. Listed in arbitrary order.

EurIPS 2025 Workshop on Principles of Generative Modeling (PriGM).

structure of the true posterior. Through experiments on flow matching, we demonstrate that introducing uncertainty on the model parameters while respecting the true posterior geometry is an effective way to reduce memorisation without forgetting how to generalise.

Specifically, our contributions are:

- We extend a method for estimating generative uncertainty as in [13] to *adapt to the geometric structure* of the model. Specifically, we define a geometry-informed approximate posterior distribution over model parameters of diffusion-like generative models.
- We provide empirical evidence that respecting the geometric structure can help generative models *generalise rather than memorise*, as our posterior predictive is based on parameter samples from high-density regions of the true posterior.

2 Methods

Notation. A neural network is a function $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^D$ and $\mathcal{Y} \in \mathbb{R}^M$ are the input and output spaces, respectively. This function depends on the model parameters $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^K$, where Θ denotes the parameter space. Using a training data set $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$ we aim to find a fixed set of parameters that minimise a loss function, i.e.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) = \arg\min_{\boldsymbol{\theta}} \; \frac{1}{N} \sum_{i=1}^{N} \ell\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i\right), \tag{1}$$

where ℓ is the per-sample loss, for example the sum of squared errors.

Bayesian deep learning. One way to quantify uncertainty of a neural network is to consider the variation in the outputs of an ensemble of learners. Instead of training multiple neural networks, a common approach is to treat the model parameters in a *Bayesian* manner by defining a posterior distribution over the model parameters using Bayes' rule. We can consider the negative log-posterior as a loss function given by

$$\mathcal{L}_{\text{post}}(\boldsymbol{\theta}, \mathcal{D}) = -\big(\log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathcal{D})\big) \tag{2}$$

$$\propto -\log p(\mathcal{D}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}). \tag{3}$$

Although the log-posterior is intractable, the fact that the log-joint is proportional to it allows us to define an approximate posterior distribution over the model parameters that respects the local geometry at the optimum. We can define this distribution *post-hoc* using the Laplace approximation:

$$q(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}^*, \boldsymbol{\Sigma}), \tag{4}$$

where $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{post}}$ is the *maximum a posteriori* (MAP) estimate and $\boldsymbol{\Sigma}^{-1} = \nabla^2_{\theta} \mathcal{L}_{\text{post}}(\boldsymbol{\theta}^*, \mathcal{D})$. For further reading we recommend the book [17].

Riemannian Laplace approximation. A recent work [1] introduced a Riemannian formulation of the Laplace approximation, which outperforms the conventional Laplace approximation by respecting the geometric structure of the true posterior. This approach constrains samples from the approximate posterior to lie within high-density regions of the true posterior, while the Euclidean Laplace approximation has no such guarantee.

We define the (posterior) loss manifold as the graph of the posterior loss function (Equation 3):

$$\mathcal{M} = \{h(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \mathcal{L}_{\text{post}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\} \in \mathbb{R}^{(K+1)}, \tag{5}$$

which yields a submanifold of $\mathbb{R}^{(K+1)}$. This parametrisation has also been considered in [1, 9, 11, 19], for instance. The Riemannian Laplace approximation $q_{\mathcal{M}}(\boldsymbol{\theta}|\mathcal{D})$ is obtained by considering a distribution of vectors $\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ in the parameter space Θ, and sampling $\boldsymbol{\theta}_s \sim q_{\mathcal{M}}(\boldsymbol{\theta}|\mathcal{D})$ as the endpoint of the geodesic starting at the MAP estimate $\boldsymbol{\theta}^*$ with initial velocity $\boldsymbol{v}$. A geodesic defines the locally shortest path on a manifold and, in mathematical terms, satisfies the *geodesic equation*:

$$\ddot{\alpha}_k(t) = -\sum_{i,j=1}^{n} \dot{\alpha}_i(t)\dot{\alpha}_j(t) \cdot \Gamma^k_{ij}(\alpha(t)), \tag{6}$$

where Γ_{ij}^k define the Christoffel symbols. In practice, we solve this second order ODE using the Runge-Kutta method [5] of order 5, subject to the initial conditions

$$\alpha(0) = \boldsymbol{\theta}^* \qquad \text{and} \qquad \dot{\alpha}(0) = \boldsymbol{v}.$$

We evaluate at $t = 1$ to obtain a sample from the Riemannian approximate posterior, $\boldsymbol{\theta}_s = \alpha(1)$. A similar approach is taken in [12] in the context of input data augmentation and we provide further details in Appendix A.

Flow matching. Assume we have N data samples $\{\boldsymbol{x}_*^i\}_{i=1}^N \in \mathcal{X} \subseteq \mathbb{R}^D$ from an unknown distribution p_*. Our goal is to find a generator function $g_{\boldsymbol{\theta}}$ that maps samples from a known base distribution $\boldsymbol{x}_0 \sim p_0$ to new samples $\hat{\boldsymbol{x}} = g_{\boldsymbol{\theta}}(\boldsymbol{x}_0)$ that approximately belong to p_*. In flow matching, the generator's output solves an initial value problem (IVP) of the form

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \qquad \dot{\boldsymbol{x}}(t) = u_\theta(\boldsymbol{x}(t), t), \tag{7}$$

where $u_{\boldsymbol{\theta}} : \mathbb{R}^D \times [0,1] \to \mathbb{R}^D$ is a velocity field, represented by a neural network with parameters $\boldsymbol{\theta}$. Usually, the base distribution is the unit Gaussian distribution $p_0 = \mathcal{N}(\mathbf{0}, \mathbb{I}_D)$ and the IVP is solved with an Euler scheme. We denote the distribution of generated samples by $\hat{p}$.

To learn the velocity field $u_{\boldsymbol{\theta}}$, we optimise the following loss function:

$$\mathcal{L}_{\text{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{t\sim\mathcal{U}_{[0,1]},\boldsymbol{x}_*\sim p_*,\boldsymbol{x}_0\sim p_0}\left[\|u_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - (\boldsymbol{x}_* - \boldsymbol{x}_0)\|_2^2\right]. \tag{8}$$

We denote a sample from the conditional probability path by $\boldsymbol{x}_t \sim p_t(\cdot|\boldsymbol{x}_*)$ and consider a Gaussian conditional optimal transport path, thus $\boldsymbol{x}_t = t\boldsymbol{x}_* + (1-t)\boldsymbol{x}_0$ for uniformly distributed time samples $t \sim \mathcal{U}_{[0,1]}$. For further reading, we recommend the lecture notes [10].

Nearest neighbour memorisation measure. We say that a point is memorised if it is much closer to one point from the training set than to all others [4, 20]. This measure is easy to compute using the Euclidean distance. A generated data sample, $\hat{\boldsymbol{x}}$, is memorised if for a fixed threshold $c \in (0,1)$,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{(1)}(\hat{\boldsymbol{x}})\|^2 \leq c\|\hat{\boldsymbol{x}} - \boldsymbol{x}^{(2)}(\hat{\boldsymbol{x}})\|^2, \tag{9}$$

where $\boldsymbol{x}^{(1)}(\hat{\boldsymbol{x}})$ and $\boldsymbol{x}^{(2)}(\hat{\boldsymbol{x}})$ are the closest and second closest training samples to $\hat{\boldsymbol{x}}$, respectively. The corresponding memorisation ratio [4] of a generator $g_{\boldsymbol{\theta}}$ is given by

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\|\hat{\boldsymbol{x}}^i - \boldsymbol{x}^{(1)}\left(\hat{\boldsymbol{x}}^i\right)\|^2 \leq c\|\hat{\boldsymbol{x}}^i - \boldsymbol{x}^{(2)}\left(\hat{\boldsymbol{x}}^i\right)\|^2\right], \tag{10}$$

where $\{\hat{\boldsymbol{x}}^i = g_{\boldsymbol{\theta}}\left(\boldsymbol{x}_0^i\right) \mid \boldsymbol{x}_0^i \sim p_0\}_{i=1}^N$ is a set of N generated data points.

3 Experiment

We consider a one-dimensional flow matching problem. Specifically, we let the true target distribution p_* be an equally weighted mixture of Gaussians (GMM) with means $\{\mu_1, \mu_2\} = \{-1.5, 1.5\}$ and variances $\sigma_1^2 = \sigma_2^2 = 0.1$. We define the base distribution as the unit Gaussian distribution $p_0 = \mathcal{N}(0,1)$. For constructing a generative model that *memorises* (i.e. overfits), we construct a naively simple training set by restricting the samples to consist of only 2 data samples: μ_1 and μ_2. We visualise the learning problem and the overfitted velocity field in Figure 2 (*left*).

We adopt a Bayesian treatment of the model parameters using the Euclidean and Riemannian Laplace approximations by setting the flow matching loss, $\mathcal{L}_{\text{FM}}$, as the log-likelihood term, $\log p(\mathcal{D}|\boldsymbol{\theta})$, and choosing a uniform prior over the weights. We sample $S = 1000$ model realisations $\boldsymbol{\theta}_s \sim q(\boldsymbol{\theta}|\mathcal{D})$, which gives $S = 1000$ different generator function $g_{\boldsymbol{\theta}_s}$ for each method. We apply these generator functions to $N = 1000$ noise samples drawn from the base distribution p_0 and analyse the memorisation ratio and generalisation of the generated data in Figure 3. We refer to Appendix B for further details on the experimental setup.

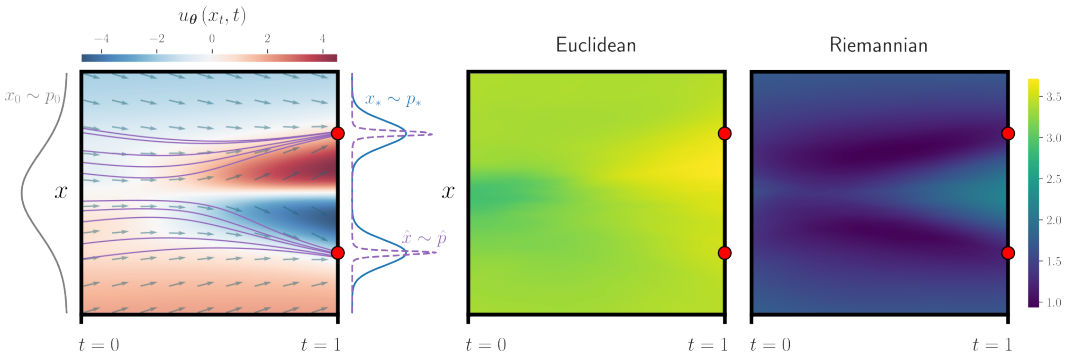

Figure 2: We consider a 1D flow matching toy problem with a Gaussian base distribution p_0 and a Gaussian mixture model (GMM) as the target distribution p_*. *Left:* the learnt conditional vector field $u_{\boldsymbol{\theta}}(x_t, t) : \mathbb{R} \times [0, 1] \to \mathbb{R}$ at the optimal parameters $\boldsymbol{\theta}^*$ for $x_t \in [-3, 3]$ and $t \in [0, 1]$. Each line (—) is a sample path obtained by following the trajectory of a noise sample $x_0 \sim p_0$ using $u_{\boldsymbol{\theta}^*}$ to form the distribution of generated samples $\hat{x} \sim \hat{p}$. The distribution of generated samples $\hat{p}$ overfits to the two fixed training samples (●) and we say that the generator $g_{\boldsymbol{\theta}^*}$ has learnt to *memorise* rather than generalise to p_*. *Right:* we consider $S = 1000$ different model realisations $\boldsymbol{\theta}_s \sim q(\boldsymbol{\theta}|\mathcal{D})$ drawn from a Euclidean and Riemannian approximate posterior, respectively. Each realisation gives a specific velocity field (similar to *left*). We show the standard deviation per (x_t, t)-coordinate computed over the S different velocity fields.

Toy example results. The results in Figure 2 indicate that the standard deviation of the velocity field is higher in the Euclidean than the Riemannian setting across the entire domain. Ignoring the local intrinsic structures of the loss manifold, the Euclidean setting treats all directions equally. The Riemannian approach on the other hand accounts for the slope and the perturbations progress more hesitantly into low-probability regions, which results in a more controlled exploration of the parameter space. The heat map reveals a clear correspondence between the underlying vector field and the Riemannian standard deviation. In regions where the vector field has low magnitude, the standard deviation is small, whereas the standard deviation increases in regions with larger magnitudes. This suggests that the Riemannian approach dynamically adapts to the underlying landscape and allows larger perturbations only when the model is confident. This is also reflected in the symmetry patterns: the learnt vector field is an odd function in x, while the Riemannian standard deviation is even, which suggests that velocity fields associated to parameter samples from the Riemannian posterior still respect the symmetry of the original learnt model. The Euclidean heat map has no such symmetry.

Figure 3 reveals that adding noise to the model parameters by sampling the approximate posterior distributions reduces memorisation, no matter the choice of distance threshold c. While data points generated with the Euclidean approximate posterior exhibit less memorisation than data points generated with the Riemannian approach and the non-Bayesian model, this reduced memorisation comes at the comes at the expense of poorer generalisation to the true underlying distribution. In contrast, the generated data distribution using the Riemannian approach still memorises less than the original generator $g_{\boldsymbol{\theta}^*}$ while generalising *better* than the other two approaches.

4 Discussion

Balancing generalisation and memorisation. Our results highlight that reducing memorisation alone does not guarantee improved generalisation; rather, we observe a delicate trade-off between memorisation and generalisation. The Euclidean method exhibits the least memorisation, suggesting that the parameter perturbation encourages a broader variance over the output space. However, this comes at the cost of poorer generalisation, demonstrating that counteracting memorisation too aggressively can prevent the model from learning meaningful structures. The Riemannian method, on the other hand, provides a good balance: it still exhibits less memorisation than the baseline, while achieving substantially stronger generalisation.

How should we measure generalisation? The nearest neighbours memorisation ratio defined in Equation 10 is easy to evaluate, however a downside is the dependence on Euclidean distance, which

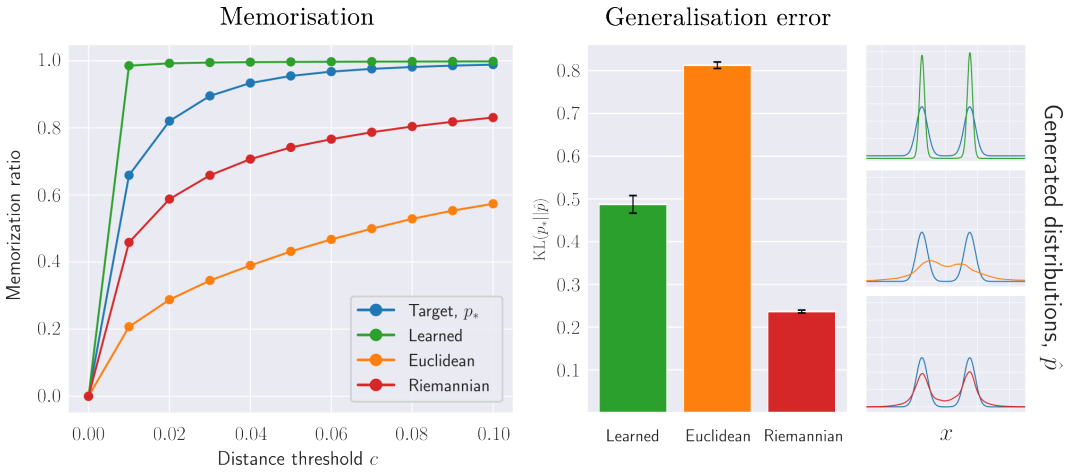

Figure 3: *Left:* The memorisation ratio (Equation 10) as a function of the distance threshold c when generating data from the target distribution, the learnt distribution and the learnt distribution under a Bayesian and Riemannian Bayesian treatment of the model parameters. *Right:* The generalisation error computed as the KL-divergence between the target distribution and the generated data distributions. For efficiency, we perform 50 repetitions of computing KL-divergences from a subset of 100 generated data samples and plot the means and standard errors in the bar plot. The generated data distributions correspond to pushing noise samples from p_0 through the generator $g_{\boldsymbol{\theta}}$ using the learnt model $\boldsymbol{\theta}^*$ (top), using several models sampled from the Euclidean Laplace approximation (middle), and using several models sampled from the Riemannian Laplace approximation (bottom). We visualise the resulting distributions using kernel density estimation. See Appendix B for details.

does not take the structure of the data manifold into account and therefore does not necessarily align with semantic similarity in the data domain. Furthermore, the choice of the constant c is somewhat arbitrary. We conclude that the nearest neighbours memorisation ratio may be more suitable for a global than a per instance memorisation measure.

Different measures for memorisation have been proposed, for example in [18]. The latter work compares the local intrinsic dimension (LID) of the ground truth data manifold and of the learnt data manifold. We say that a point is memorised if the dimension of the learnt manifold is lower than the dimension of the ground truth manifold. This measure carries a nice geometric flavour and provides more qualitative information about the structure of the learnt distribution than merely evaluating distances between points. However, the estimation of the LID is computationally heavy and harder to evaluate, although improvements have been made in [15].

Conclusion, limitations and future work. As we have demonstrated empirically, adopting a geometry-informed Bayesian treatment of the model parameters can help reduce memorisation, without loosing generalisation capabilities. This finding is based on a simple one-dimensional example with only two training points being the means of a 2-component mixture of Gaussians. This allowed for visually interpreting the effects of our Riemannian Bayesian treatment of the model parameters, however recent work [3] suggests that the local dynamical structure of diffusion models and dimension of the data sets impacts memorisation and that the dataset must be exponentially large in input space dimension to not be memorised. In future revisions, we plan to extend the experimental part to more complex toy experiments as well as real datasets that exhibit meaningful manifold structure. Additionally, we will provide the associated theoretical justification for our proposed design, and larger-scale experiments on generative image models.

Acknowledgments and Disclosure of Funding

This work was supported by Danish Data Science Academy, which is funded by the Novo Nordisk Foundation (NNF21SA0069429) and VILLUM FONDEN (40516), and by the DFF Sapere Aude Starting Grant "Geometric Analysis of Deep Learning".

References

[1] Federico Bergamin, Pablo Moreno-Muñoz, Søren Hauberg, and Georgios Arvanitidis. Riemannian Laplace approximations for Bayesian neural networks. *Neural Information Processing Systems (NeurIPS)*, 2023.

[2] Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. *Neural Information Processing Systems (NeurIPS)*, 2025.

[3] Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

[4] Sam Buchanan, Druv Pai, Yi Ma, and Valentin De Bortoli. On the edge of memorization in diffusion models. *Neural Information Processing Systems (NeurIPS)*, 2025.

[5] John R Dormand and Peter J Prince. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[6] Tyler Farghly, Peter Potaptchik, Samuel Howard, George Deligiannidis, and Jakiw Pidstrigach. Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive. *Neural Information Processing Systems (NeurIPS)*, 2025.

[7] Tyler Farghly, Patrick Rebeschini, George Deligiannidis, and Arnaud Doucet. Implicit regularisation in diffusion models: An algorithm-dependent generalisation analysis. *arXiv:2507.03756*, 2025.

[8] Sigmundur Gudmundsson. *An Introduction to Riemannian Geometry*. 2025.

[9] Marcelo Hartmann, Mark Girolami, and Arto Klami. Lagrangian manifold Monte Carlo on Monge patches. *International Conference on Artificial Intelligence and Statistics*, 2022.

[10] Peter Holderrieth and Ezra Erives. Introduction to flow matching and diffusion models, 2025.

[11] Albert Kjøller Jacobsen and Georgios Arvanitidis. Monge SAM: Robust reparameterization-invariant sharpness-aware minimization based on loss geometry. *arXiv:2502.08448*, 2025.

[12] Albert Kjøller Jacobsen, Johanna Marie Gegenfurtner, and Georgios Arvanitidis. Staying on the manifold: Geometry-aware noise injection. *Northern Lights Deep Learning Conference (NLDL)*, 2026.

[13] Metod Jazbec, Eliot Wong-Toi, Guoxuan Xia, Dan Zhang, Eric Nalisnick, and Stephan Mandt. Generative uncertainty in diffusion models. *Uncertainty in Artificial Intelligence (UAI)*, 2025.

[14] Dongjae Jeon, Dueun Kim, and Albert No. Understanding and mitigating memorization in generative models via sharpness of probability landscapes. *International Conference on Machine Learning (ICML)*, 2024.

[15] Hamidreza Kamkari, Brendan Leigh Ross, Rasa Hosseinzadeh, Jesse C. Cresswell, and Gabriel Loaiza-Ganem. A geometric view of data complexity: Efficient local intrinsic dimension estimation with diffusion models. *arxiv:2406.03537*, 2024.

[16] John M. Lee. *Introduction to Smooth Manifolds*. 2000.

[17] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[18] Brendan Leigh Ross, Hamidreza Kamkari, Tongzi Wu, Rasa Hosseinzadeh, Zhaoyan Liu, George Stein, Jesse C Cresswell, and Gabriel Loaiza-Ganem. A geometric framework for understanding memorization in generative models. *Neural Information Processing Systems (NeurIPS)*, 2024.

[19] Bernardo Williams, Hanlin Yu, Hoang Phuc Hau Luu, Georgios Arvanitidis, and Arto Klami. Geodesic slice sampler for multimodal distributions with strong curvature. *Uncertainty in Artificial Intelligence (UAI)*, 2025.

[20] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 workshop on structured probabilistic inference & generative modeling*, 2023.

[21] Hanlin Yu, Marcelo Hartmann, Bernardo Williams, Mark Girolami, and Arto Klami. Riemannian Laplace approximation with the Fisher metric. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

A A brief excursion to differential geometry.

The (posterior) loss manifold is the graph of the posterior loss function 3:

$$\mathcal{M} = \{h(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \mathcal{L}_{\text{post}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\} \in \mathbb{R}^{(K+1)}.$$

Here,

$$h : \Theta \to \mathbb{R}^{K+1}, \quad h(\boldsymbol{\theta}) = (\boldsymbol{\theta}, \mathcal{L}_{\text{post}}(\boldsymbol{\theta}))$$

is the parametrisation of $\mathcal{M}$ induced by $\mathcal{L}_{\text{post}}$.

We equip $\mathcal{M}$ with the *pullback* metric as follows. Consider two smooth curves

$$\alpha_1, \alpha_2 : [0,1] \to \Theta$$

in the parameter space, such that $\alpha_1(0) = \alpha_2(0) = \boldsymbol{\theta}$. Then

$$\gamma_1 = h \circ \alpha_1, \gamma_2 = h \circ \alpha_2 : [0,1] \to \mathcal{M}$$

are smooth curves on $\mathcal{M}$, intersecting at $p = h(\boldsymbol{\theta})$. We evaluate the scalar product at $p \in \mathcal{M}$ by computing

$$\begin{aligned} \langle \dot{\alpha_1}(0), \dot{\alpha_2}(0) \rangle_{\mathbf{G}} &= \dot{\alpha_1}(0) \mathbf{G}(\boldsymbol{\theta}) \dot{\alpha_2}(0)^\top \\ &= \langle \dot{\gamma_1}(0), \dot{\gamma_2}(0) \rangle, \end{aligned}$$

where the matrix $\mathbf{G}(\boldsymbol{\theta})$ is given by

$$\mathbf{G}(\boldsymbol{\theta}) = \nabla_\theta h \nabla_\theta h^\top = \mathbb{I}_K + \nabla_\theta \mathcal{L}(\boldsymbol{\theta}) \nabla_\theta \mathcal{L}(\boldsymbol{\theta})^\top. \tag{11}$$

We refer to locally shortest paths on a manifold as geodesics. A curve $\gamma(t) = h \circ \alpha(t)$ on the manifold is a geodesic if and only if $\alpha(t)$ satisfies the geodesic equation:

$$\ddot{\alpha}_k(t) = -\sum_{i,j=1}^{n} \dot{\alpha}_i(t) \dot{\alpha}_j(t) \cdot \Gamma_{ij}^k(\alpha(t)). \tag{12}$$

Here, Γ_{ij}^k define the Christoffel symbols. If $\mathcal{M}$ is *regular* and *complete*, then for each point $p \in \mathcal{M}$ and each unit tangent vector $v \in T_p\mathcal{M}$ in the tangent space at p, there exists precisely one geodesic through p in the direction of v. For further details, we refer the reader to the classic textbook in differential geometry [16] or the lecture notes [8].

B Experimental details

We adopt a Bayesian treatment of the parameters of $u_{\boldsymbol{\theta}}$, by defining the likelihood term in Equation 3 as the flow matching loss $\mathcal{L}_{\text{FM}}(\boldsymbol{\theta})$ and use a uniform prior over the weights. This corresponds to placing the Laplace approximation at the maximum likelihood estimate found from training the flow model with gradient descent. We consider the dataset $\mathcal{D}$ as a fixed collection of data-noise pairings and equidistant time samples, i.e. $\mathcal{D} = \{t_i = \frac{i}{N}, \boldsymbol{x}_0^i, \boldsymbol{x}_*^i\}_{n=1}^N$, for ensuring a deterministic loss.

We sample $S = 1000$ model realisations from the Euclidean approximate posterior $\boldsymbol{\theta}_{\text{E-LA}}^s \sim q(\boldsymbol{\theta}|\mathcal{D})$ and additionally use these as initial velocities for obtaining $S = 1000$ samples from the Riemannian approximate posterior $\boldsymbol{\theta}_{\text{R-LA}}^s \sim q_{\mathcal{M}}(\boldsymbol{\theta}|\mathcal{D})$. For both approximate posterior methods, we compute the velocity field over a space-time grid for each of the model realisations $\boldsymbol{\theta}_s$. We visualise the variation over the associated velocity fields in Figure 2 (*right*).

Next, we sample $N = 1000$ points from the base distribution and push these through the generator associated to every model realisation. This gives us a set of $N \times S$ generated points for each approximate posterior. We formalise this as:

$$\{\hat{x}^i_{\text{E-LA}}\}_{i=1}^{N\times S} = \{\, g_{\boldsymbol{\theta}_s}(x_0^i) \mid x_0^i \sim p_0,\, \boldsymbol{\theta}^s_{\text{E-LA}} \sim q\left(\boldsymbol{\theta}|\mathcal{D}\right) \}_{i=1,s=1}^{N,S} \tag{13}$$

$$\{\hat{x}^i_{\text{R-LA}}\}_{i=1}^{N\times S} = \{\, g_{\boldsymbol{\theta}_s}(x_0^i) \mid x_0^i \sim p_0,\, \boldsymbol{\theta}^s_{\text{R-LA}} \sim q_{\mathcal{M}}\left(\boldsymbol{\theta}|\mathcal{D}\right) \}_{i=1,s=1}^{N,S} \tag{14}$$

and visualise the memorisation ratio for various distance thresholds c in Figure 3 along with the distributions of generated samples $(\hat{p},\ \hat{p}_{\text{E-LA}},\ \hat{p}_{\text{R-LA}})$ and report the KL-divergence to the true target distribution p_* as a measure of generalisation.