



# SBGM: Score-Based Generative Models in JAX.

Jed Homer <sup>1,2\*</sup>

<sup>1</sup> University Observatory, Faculty for Physics, Ludwig-Maximilians-Universität München, Scheinerstrasse 1, München, Deutschland.  <sup>2</sup> Munich Center for Machine Learning.  \* These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

## Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## In partnership with



AMERICAN  
ASTRONOMICAL  
SOCIETY

This article and software are linked with research article DOI [10.3847/xxxxx](https://doi.org/10.3847/xxxxx) <- [update this with the DOI from AAS once you know it.](#), published in the [Astrophysical Journal](#) <- The name of the AAS journal..

## Summary

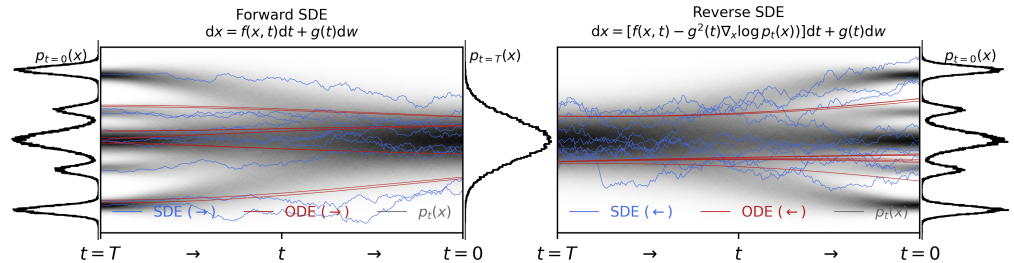
Diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song, Sohl-Dickstein, et al., 2021) have emerged as the dominant paradigm for generative modelling based on performance at a variety of tasks (Peebles & Xie, 2023; Rombach et al., 2022). The advantages of accurate density estimation and high-quality samples of normalising flows (Grathwohl et al., 2018; Papamakarios et al., 2021), VAEs (Diederik P. Kingma & Welling, 2022) and GANs (Goodfellow et al., 2014) are subsumed into this method. Significant limitations exist on implicit and neural network based likelihood models with respect to modelling normalised probability distributions and sampling speed. Score-matching diffusion models are more efficient than previous generative model algorithms for these tasks. The diffusion process is agnostic to the data representation meaning different types of data such as audio, point-clouds, videos and images can be modelled. The use of generative models, such as diffusion models, remains somewhat unexplored given the amount of research into these methods in the machine learning community. In order to bridge the gap, trusted software is needed to allow research in the natural sciences using generative models.

## Statement of need

Diffusion-based generative models (Ho et al., 2020; Sohl-Dickstein et al., 2015) are a method for sampling from high-dimensional distributions. A sub-class of these models, score-based diffusion generative models (SBGMs, (Song, Sohl-Dickstein, et al., 2021)), permit exact-likelihood estimation via a change-of-variables associated with the forward diffusion process (Song, Durkan, et al., 2021). Diffusion models allow fitting generative models to high-dimensional data in a more efficient way than normalising flows since only one neural network model parameterises the diffusion process as opposed to a sequence of neural networks in typical normalising flow architectures. Whilst existing diffusion models (Ho et al., 2020; Diederik P. Kingma et al., 2023) allow for sampling, they are limited to inaccurate variational inference approaches for density estimation which limits their use for Bayesian inference. This code provides density estimation with diffusion models using GPU enabled ODE solvers in `jax` (Bradbury et al., 2018) and `diffrax` (Kidger, 2022).

The software we present, `sbgm`, is designed to be used by researchers in machine learning and the natural sciences for fitting diffusion models with a suite of custom architectures for their research. These models can be fit easily with multi-accelerator training and inference within the code. Typical use cases for these kinds of generative models are emulator approaches (Spurio Mancini et al., 2022), simulation-based inference (Cranmer et al., 2020), field-level inference (Andrews et al., 2023) and general inverse problems (Feng et al., 2023; Feng & Bouman, 2024; Remy et al., 2023; Song et al., 2022) (e.g. image inpainting (Song, Sohl-Dickstein, et al., 2021) and denoising (Chung et al., 2022; Daras et al., 2024)). This code allows for seamless integration of diffusion models to these applications by providing data-generating models

with easy conditioning of the data on parameters, classifying variables or other data such as images. Furthermore, the implementation in equinox (Kidger & Garcia, 2021) guarantees safe integration of sbgm with any other sampling libraries (e.g. BlackJAX Cabezas et al. (2024)) or jax (Bradbury et al., 2018) based codes.



**Figure 1:** A diagram showing how to map data to a noise distribution (the prior) with an SDE, and reverse this SDE for generative modeling. One can also reverse the associated probability flow ODE, which yields a deterministic process that samples from the same distribution as the SDE. Both the reverse-time SDE and probability flow ODE can be obtained by estimating the score.

## Diffusion

Diffusion in the context of generative modelling describes the process of adding small amounts of noise sequentially to samples of data  $x$  (Sohl-Dickstein et al., 2015). A generative model for the data arises from training a neural network to reverse this process by subtracting the noise added to the data.

We assume a data distribution  $q(x_0)$  and a continuous sequence of increasing noise levels  $\sigma_t(t)$  as a function of the diffusion time  $t$ . The data is perturbed by Gaussian noise

Score-based diffusion models (Song, Sohl-Dickstein, et al., 2021) model the forward diffusion process with Stochastic Differential Equations (SDEs) of the form

$$dx = f(x, t)dt + g(t)dw,$$

where  $f(x, t)$  is a vector-valued function called the drift coefficient,  $g(t)$  is the diffusion coefficient and  $dw$  is a sample of noise  $dw \sim \mathcal{G}[dw|0, I]$ . This equation describes the infinitely many samples of noise along the diffusion time  $t$  that perturb the data. The diffusion path begins at  $t = 0$  and ends at  $T = 0$  where the resulting distribution is then a multivariate Gaussian with mean zero and covariance  $I$ .

The SDE itself is parameterised by  $t$  and  $x$  and existing options include the variance exploding (VE), variance preserving (VP) and sub-variance preserving (SubVP).

The reverse of the SDE, mapping from multivariate Gaussian samples  $x(T)$  to samples of data  $x(0)$ , is of the form

$$dx = [f(x, t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dw,$$

where the score function  $\nabla_x \log p_t(x)$  is substituted with a neural network  $s_\theta(x(t), t)$  for the sampling process. The network is fit by score-matching (Hyvärinen, 2005; Vincent, 2011) across the time span  $[0, T]$ . This network predicts the noise added to the image at time  $t$  with the forward diffusion process, in accordance with the SDE, and removes it. With a data-dimensional sample of Gaussian noise from the prior  $p_T(x)$  (see Figure 1) one can reverse the diffusion process to generate data.

71 In Figure 1 the forward and reverse diffusion processes are shown for a samples from a  
72 one-dimensional mixture of Gaussians with their corresponding SDE and ODE paths.

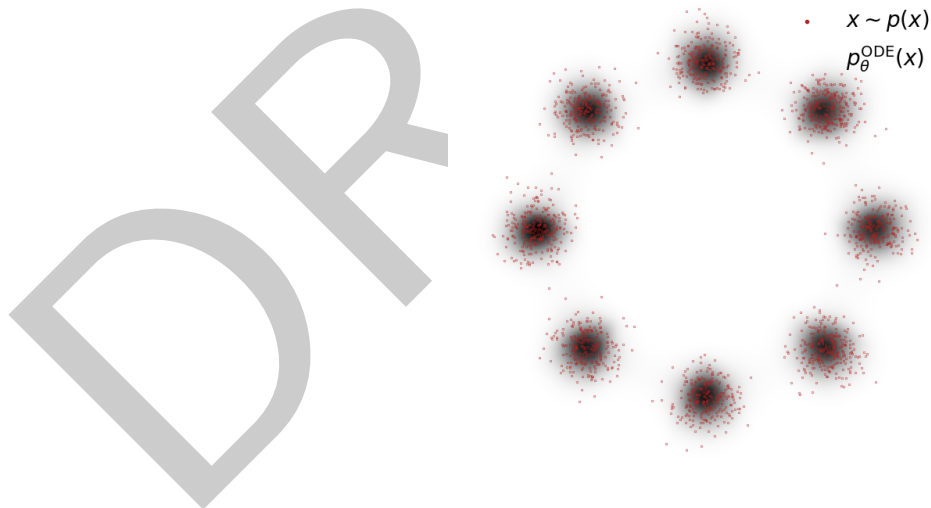
73 The reverse SDE may be solved with Euler-Murayama sampling (Song, Sohl-Dickstein, et al.,  
74 2021) (or other annealed Langevin sampling methods) which is featured in the code.

## 75 Likelihood calculations with diffusion models

76 Many of the applications of generative models depend on being able to calculate the likelihood  
77 of data. In Song, Sohl-Dickstein, et al. (2021) it is shown that any SDE may be converted  
78 into an ordinary differential equation (ODE) without changing the distributions, defined by the  
79 SDE, from which the noise is sampled from in the diffusion process (denoted  $p_t(x)$  and shown  
80 in grey in Figure 1). This ODE is known as the probability flow ODE (Song, Sohl-Dickstein,  
81 et al., 2021; Song, Durkan, et al., 2021) and is written

$$dx = [f(x, t) - g^2(t)\nabla_x \log p_t(x)]dt = f'(x, t)dt.$$

82 This ODE can be solved with an initial-value problem. Starting with a data point  $x(0) \sim p(x)$ ,  
83 this point is mapped along the probability flow ODE path (see the right-hand side of Figure 1)  
84 to a sample from the multivariate Gaussian prior. This inherits the formalism of continuous  
85 normalising flows (Chen et al., 2019; Grathwohl et al., 2018) without the expensive ODE  
86 simulations used to train these models - allowing for a likelihood estimate based on diffusion  
87 models (Song, Durkan, et al., 2021). The initial value problem provides a solution  $x(T)$  and  
88 the change in probability along the path  $\Delta = \log p(x(0)) - \log p(x(T))$  where  $p(x(T))$  is a  
89 simple multivariate Gaussian distribution.



**Figure 2:** A diagram showing a log-likelihood calculation over the support of a Gaussian mixture model with eight components. Data is drawn (shown in red) from this mixture to train the diffusion model that gives the likelihood in gray. The log-likelihood is calculated using the ODE and a trained diffusion model.

90 The likelihood estimate under a score-based diffusion model is estimated by solving the  
91 change-of-variables equation for continuous normalising flows.

$$\frac{\partial}{\partial t} \log p(x(t)) = \nabla_x \cdot f(x(t), t),$$

92 which gives the log-likelihood of a single datapoint  $x(0)$  as

$$\log p(x(0)) = \log p(x(T)) + \int_{t=0}^{t=T} dt \nabla_x \cdot f(x, t).$$

93 The code implements these calculations also for the Hutchinson trace estimation method  
 94 (Grathwohl et al., 2018) that reduces the computational expense of the estimate. Figure 2  
 95 shows an example of a data-likelihood calculation using a trained diffusion model with the  
 96 ODE associated from an SDE. It is possible to train score-based diffusion models such that  
 97 the score-matching loss bounds the Kullback-Leibler divergence for each data point  $x$  against  
 98 the unknown data distribution (shown in (Song, Durkan, et al., 2021)). It is also implemented  
 99 in the code such that the score-matching bounds the KL divergence between the model and  
 100 unknown data distribution per datapoint.

## 101 Implementations and future work

102 Diffusion models are defined in sbgm via a score-network model  $s_\theta$  and an SDE. All the  
 103 available SDEs in the literature of score-based diffusion models are available. We provide  
 104 implementations for UNet (Ronneberger et al., 2015), MLP-Mixer (Tolstikhin et al., 2021) and  
 105 Residual Network (He et al., 2015) models which are state-of-the-art for diffusion tasks. It is  
 106 possible to fit score-based diffusion models to a conditional distribution  $p(x|\pi, y)$  where in  
 107 typical inverse problems  $y$  would be an image and  $\pi$  a set of parameters in a physical model for  
 108 the data (?) (e.g. to solve inverse problems). The code is compatible with any model written  
 109 in the equinox (Kidger & Garcia, 2021) framework. We are extending the code to provide  
 110 transformer-based (Peebles & Xie, 2023) and latent diffusion models (Rombach et al., 2022).

## 111 GPU Support

112 sbgm offers easy GPU support including the use of multiple GPU devices for training and  
 113 sampling within the code.

## 114 Acknowledgements

115 We thank the developers of the packages jax (Bradbury et al., 2018), optax (DeepMind et  
 116 al., 2020), equinox (Kidger & Garcia, 2021) and diffrax (Kidger, 2022) for their work and  
 117 for making their code available to the community.

## 118 References

- 119 Andrews, A., Jasche, J., Lavaux, G., & Schmidt, F. (2023). Bayesian field-level inference of  
 120 primordial non-gaussianity using next-generation galaxy surveys. *Monthly Notices of the*  
 121 *Royal Astronomical Society*, 520(4), 5746–5763. <https://doi.org/10.1093/mnras/stad432>
- 122 Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G.,  
 123 Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable*  
 124 *transformations of Python+NumPy programs* (Version 0.3.13). [http://github.com/jax-ml/](http://github.com/jax-ml/jax)  
 125 [jax](http://github.com/jax-ml/jax)
- 126 Cabezas, A., Corenflos, A., Lao, J., & Louf, R. (2024). *BlackJAX: Composable Bayesian*  
 127 *inference in JAX*. <https://arxiv.org/abs/2402.10797>
- 128 Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2019). *Neural ordinary*  
 129 *differential equations*. <https://arxiv.org/abs/1806.07366>

- 130 Chung, H., Kim, J., Kim, S., & Ye, J. C. (2022). *Parallel diffusion models of operator and*  
 131 *image for blind inverse problems.* <https://arxiv.org/abs/2211.10656>
- 132 Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference.  
 133 *Proceedings of the National Academy of Sciences*, 117(48), 30055–30062. <https://doi.org/10.1073/pnas.1912789117>
- 134
- 135 Daras, G., Dimakis, A. G., & Daskalakis, C. (2024). *Consistent diffusion meets tweedie:*  
 136 *Training exact ambient diffusion models with noisy data.* <https://arxiv.org/abs/2404.10177>
- 137 DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P.,  
 138 Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones,  
 139 C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., ... Viola, F. (2020). *The DeepMind*  
 140 *JAX Ecosystem.* <http://github.com/google-deepmind>
- 141 Feng, B. T., & Bouman, K. L. (2024). *Variational bayesian imaging with an efficient surrogate*  
 142 *score-based prior.* <https://arxiv.org/abs/2309.01949>
- 143 Feng, B. T., Smith, J., Rubinstein, M., Chang, H., Bouman, K. L., & Freeman, W. T.  
 144 (2023). *Score-based diffusion models as principled priors for inverse imaging.* <https://arxiv.org/abs/2304.11751>
- 145
- 146 Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville,  
 147 A., & Bengio, Y. (2014). *Generative adversarial networks.* <https://arxiv.org/abs/1406.2661>
- 148 Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., & Duvenaud, D. (2018).  
 149 *FFJORD: Free-form continuous dynamics for scalable reversible generative models.* <https://arxiv.org/abs/1810.01367>
- 150
- 151 He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition.*  
 152 <https://arxiv.org/abs/1512.03385>
- 153 Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models.* <https://arxiv.org/abs/2006.11239>
- 154
- 155 Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching.  
 156 *Journal of Machine Learning Research*, 6(24), 695–709. <http://jmlr.org/papers/v6/hyvarinen05a.html>
- 157
- 158 Kidger, P. (2022). *On neural differential equations.* <https://arxiv.org/abs/2202.02435>
- 159 Kidger, P., & Garcia, C. (2021). Equinox: Neural networks in JAX via callable PyTrees and  
 160 filtered transformations. *Differentiable Programming Workshop at Neural Information*  
 161 *Processing Systems 2021.*
- 162 Kingma, Diederik P., Salimans, T., Poole, B., & Ho, J. (2023). *Variational diffusion models.*  
 163 <https://arxiv.org/abs/2107.00630>
- 164 Kingma, Diederik P., & Welling, M. (2022). *Auto-encoding variational bayes.* <https://arxiv.org/abs/1312.6114>
- 165
- 166 Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B.  
 167 (2021). *Normalizing flows for probabilistic modeling and inference.* <https://arxiv.org/abs/1912.02762>
- 168
- 169 Peebles, W., & Xie, S. (2023). *Scalable diffusion models with transformers.* <https://arxiv.org/abs/2212.09748>
- 170
- 171 Remy, B., Lanusse, F., Jeffrey, N., Liu, J., Starck, J.-L., Osato, K., & Schrabback, T. (2023).  
 172 Probabilistic mass-mapping with neural score estimation. *Astronomy & Astrophysics*,  
 173 672, A51. <https://doi.org/10.1051/0004-6361/202243054>
- 174 Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-resolution*  
 175 *image synthesis with latent diffusion models.* <https://arxiv.org/abs/2112.10752>

- 176 Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical*  
177 *image segmentation*. <https://arxiv.org/abs/1505.04597>
- 178 Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). *Deep unsuper-*  
179 *vised learning using nonequilibrium thermodynamics*. <https://arxiv.org/abs/1503.03585>
- 180 Song, Y., Durkan, C., Murray, I., & Ermon, S. (2021). *Maximum likelihood training of*  
181 *score-based diffusion models*. <https://arxiv.org/abs/2101.09258>
- 182 Song, Y., Shen, L., Xing, L., & Ermon, S. (2022). *Solving inverse problems in medical imaging*  
183 *with score-based generative models*. <https://arxiv.org/abs/2111.08005>
- 184 Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021).  
185 *Score-based generative modeling through stochastic differential equations*. [https://arxiv.](https://arxiv.org/abs/2011.13456)  
186 [org/abs/2011.13456](https://arxiv.org/abs/2011.13456)
- 187 Spurio Mancini, A., Piras, D., Alsing, J., Joachimi, B., & Hobson, M. P. (2022). *<Scp>Cos-*  
188 *moPower</scp>: Emulating cosmological power spectra for accelerated bayesian inference*  
189 *from next-generation surveys*. *Monthly Notices of the Royal Astronomical Society*, 511(2),  
190 1771–1788. <https://doi.org/10.1093/mnras/stac064>
- 191 Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J.,  
192 Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., & Dosovitskiy, A. (2021). *MLP-mixer:*  
193 *An all-MLP architecture for vision*. <https://arxiv.org/abs/2105.01601>
- 194 Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural*  
195 *Computation*, 23(7), 1661–1674. [https://doi.org/10.1162/NECO\\_a\\_00142](https://doi.org/10.1162/NECO_a_00142)