





# SBGM: Score-Based Generative Models in JAX.

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

<sup>1</sup> Ludwig-Maximilians-Universität München, Faculty for Physics, University Observatory, München, Deutschland.  \* 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Published: unpublished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

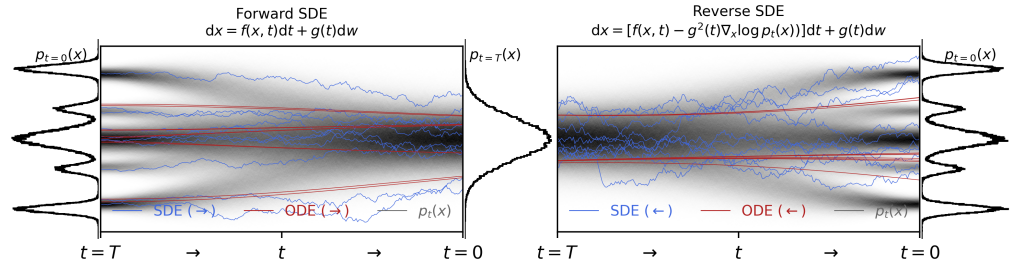
Diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song, Sohl-Dickstein, et al., 2021) have emerged as the dominant paradigm for generative modelling. The advantages of accurate density estimation and high-quality samples of normalising flows (Grathwohl et al., 2018; Papamakarios et al., 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 density estimation and sampling from high-dimensional distributions. A sub-class of these models, score-based diffusion generatives 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 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 tasks. 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 (e.g. likelihood-free inference, (?)), 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.

Score-based diffusion models 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  $dw \sim \mathcal{G}[dw|0, I]$ . The solution of a SDE is a collection of continuous random variables describing a path parameterised by a ‘time’ variable  $t$ . 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 formulated by design and existing options include the variance exploding (VE), variance preserving (VP) and sub-variance preserving (SubVP). These equations describe how the mean and covariances of the distributions of noise added to the data evolve with time.

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. 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 1) one can reverse the diffusion process to generate data.

The score-based diffusion model for the data is fit by optimising the parameters of the network  $\theta$  via stochastic gradient descent of the score-matching loss (Song, Sohl-Dickstein, et al., 2021)

$$\mathcal{L}(\theta) = \mathbf{E}_{t \sim \mathcal{U}(0, T)} \mathbf{E}_{x \sim p(x)} \mathbf{E}_{x(t) \sim p(x(t)|x)} [\lambda(t) \|\nabla_x \log p_t(x(t)|x(0)) - s_\theta(x(t), t)\|_2^2]$$

where  $\lambda(t)$  is an arbitrary scalar weighting function, chosen to weight certain times - usually near  $t = 0$  where the data has only a small amount of noise added. Here,  $p_t(x(t)|x(0))$  is the transition kernel for Gaussian diffusion paths. This is defined depending on the form of the SDE (Song, Sohl-Dickstein, et al., 2021) and for the common variance-preserving (VP) SDE the kernel is written as

$$p(x(t)|x(0)) = \mathcal{G}[x(t)|\mu_t \cdot x(0), \sigma_t^2 \cdot \mathbf{I}]$$

where  $\mathcal{G}[\cdot]$  is a Gaussian distribution,  $\mu_t = \exp(-\int_0^t \beta(s) ds)$  and  $\sigma_t^2 = 1 - \mu_t$ .  $\beta(t)$  is typically chosen to be a simple linear function of  $t$ .

In Figure 1 the forward and reverse diffusion processes are shown for a samples from a Gaussian mixture with their corresponding SDE and ODE paths.

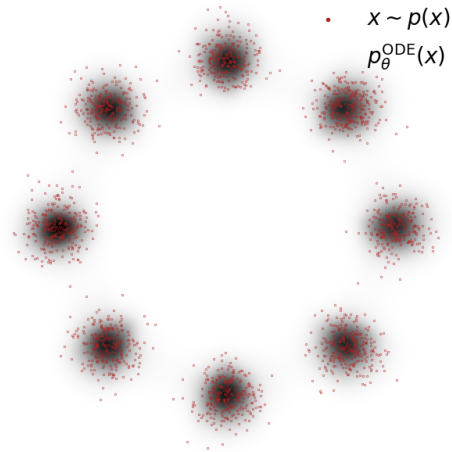
The reverse SDE may be solved with Euler-Murayama sampling (Song, Sohl-Dickstein, et al., 2021) (or other annealed Langevin sampling methods) which is featured in the code.

## Likelihood

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

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

This ODE can be solved with an initial-value problem that maps a prior sample from a multivariate Gaussian to the data distribution. This inherits the formalism of continuous normalising flows (Chen et al., 2019; Grathwohl et al., 2018) without the expensive ODE simulations used to train these flows - this allows for a likelihood estimate based on diffusion models (Song, Durkan, et al., 2021).



**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.

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

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

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

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

95 The code implements these calculations also for the Hutchinson trace estimation method  
96 (Grathwohl et al., 2018) that reduces the computational expense of the estimate. Figure 2  
97 shows an example of a data-likelihood calculation using a trained diffusion model with the  
98 ODE associated from an SDE. It is possible to train score-based diffusion models such that  
99 the score-matching loss bounds the Kullback-Leibler divergence for each data point  $x$  against  
100 the unknown data distribution. This is shown in (Song, Durkan, et al., 2021) via a choice  
101 of  $\lambda(t)$  termed the 'likelihood weighting'. It is also implemented in the code such that the  
102 score-matching bounds the KL divergence between the model and unknown data distribution  
103 per datapoint.

## 104 Conditioning diffusion models

105 As shown in (?) it is possible to fit score-based diffusion models to a conditional distribution  
106  $p(x|\pi, y)$  where in typical inverse problems  $y$  would be an image and  $\pi$  a set of parameters in  
107 a physical model for the data. The code is implemented such that all training, sampling and  
108 density estimation is possible with these inputs. This allows for diffusion models to be used  
109 many different kinds of inverse problems.

## Implementations and future work

Diffusion models are defined in sbgm via a score-network model  $s_\theta$  and an SDE. All the available SDEs in the literature of score-based diffusion models are available. We provide implementations for UNet (Ronneberger et al., 2015) and Residual Network (He et al., 2015) models which are state-of-the-art for diffusion tasks. The code is compatible with any model written in the equinox (Kidger & Garcia, 2021) framework. We are extending the code to provide transformer-based (Peebles & Xie, 2023) and latent diffusion models (Rombach et al., 2022).

Our implementation allows for the organisation of projects based on save/load configuration files, model and optimiser checkpointing and utility functions for plotting and saving metrics and sampled data.

## GPU Support

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

## Acknowledgements

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

## References

- Andrews, A., Jasche, J., Lavaux, G., & Schmidt, F. (2023). Bayesian field-level inference of primordial non-gaussianity using next-generation galaxy surveys. *Monthly Notices of the Royal Astronomical Society*, 520(4), 5746–5763. <https://doi.org/10.1093/mnras/stad432>
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). <http://github.com/jax-ml/jax>
- Cabezas, A., Corenflos, A., Lao, J., & Louf, R. (2024). *BlackJAX: Composable Bayesian inference in JAX*. <https://arxiv.org/abs/2402.10797>
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2019). *Neural ordinary differential equations*. <https://arxiv.org/abs/1806.07366>
- Chung, H., Kim, J., Kim, S., & Ye, J. C. (2022). *Parallel diffusion models of operator and image for blind inverse problems*. <https://arxiv.org/abs/2211.10656>
- Daras, G., Dimakis, A. G., & Daskalakis, C. (2024). *Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data*. <https://arxiv.org/abs/2404.10177>
- DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., ... Viola, F. (2020). *The DeepMind JAX Ecosystem*. <http://github.com/google-deepmind>
- Feng, B. T., & Bouman, K. L. (2024). *Variational bayesian imaging with an efficient surrogate score-based prior*. <https://arxiv.org/abs/2309.01949>

- 150 Feng, B. T., Smith, J., Rubinstein, M., Chang, H., Bouman, K. L., & Freeman, W. T.  
151 (2023). *Score-based diffusion models as principled priors for inverse imaging*. <https://arxiv.org/abs/2304.11751>  
152
- 153 Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville,  
154 A., & Bengio, Y. (2014). *Generative adversarial networks*. <https://arxiv.org/abs/1406.2661>
- 155 Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., & Duvenaud, D. (2018).  
156 *FFJORD: Free-form continuous dynamics for scalable reversible generative models*. <https://arxiv.org/abs/1810.01367>  
157
- 158 He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*.  
159 <https://arxiv.org/abs/1512.03385>
- 160 Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models*. <https://arxiv.org/abs/2006.11239>  
161
- 162 Kidger, P. (2022). *On neural differential equations*. <https://arxiv.org/abs/2202.02435>
- 163 Kidger, P., & Garcia, C. (2021). Equinox: Neural networks in JAX via callable PyTrees and  
164 filtered transformations. *Differentiable Programming Workshop at Neural Information*  
165 *Processing Systems 2021*.
- 166 Kingma, Diederik P., Salimans, T., Poole, B., & Ho, J. (2023). *Variational diffusion models*.  
167 <https://arxiv.org/abs/2107.00630>
- 168 Kingma, Diederik P., & Welling, M. (2022). *Auto-encoding variational bayes*. <https://arxiv.org/abs/1312.6114>  
169
- 170 Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B.  
171 (2021). *Normalizing flows for probabilistic modeling and inference*. <https://arxiv.org/abs/1912.02762>  
172
- 173 Peebles, W., & Xie, S. (2023). *Scalable diffusion models with transformers*. <https://arxiv.org/abs/2212.09748>  
174
- 175 Remy, B., Lanusse, F., Jeffrey, N., Liu, J., Starck, J.-L., Osato, K., & Schrabback, T. (2023).  
176 Probabilistic mass-mapping with neural score estimation. *Astronomy & Astrophysics*,  
177 672, A51. <https://doi.org/10.1051/0004-6361/202243054>
- 178 Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-resolution*  
179 *image synthesis with latent diffusion models*. <https://arxiv.org/abs/2112.10752>
- 180 Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical*  
181 *image segmentation*. <https://arxiv.org/abs/1505.04597>
- 182 Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). *Deep unsuper-*  
183 *vised learning using nonequilibrium thermodynamics*. <https://arxiv.org/abs/1503.03585>
- 184 Song, Y., Durkan, C., Murray, I., & Ermon, S. (2021). *Maximum likelihood training of*  
185 *score-based diffusion models*. <https://arxiv.org/abs/2101.09258>
- 186 Song, Y., Shen, L., Xing, L., & Ermon, S. (2022). *Solving inverse problems in medical imaging*  
187 *with score-based generative models*. <https://arxiv.org/abs/2111.08005>
- 188 Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021).  
189 *Score-based generative modeling through stochastic differential equations*. <https://arxiv.org/abs/2011.13456>  
190
- 191 Spurio Mancini, A., Piras, D., Alsing, J., Joachimi, B., & Hobson, M. P. (2022). *<Scp>Cos-*  
192 *moPower</scp>: Emulating cosmological power spectra for accelerated bayesian inference*  
193 *from next-generation surveys*. *Monthly Notices of the Royal Astronomical Society*, 511(2),  
194 1771–1788. <https://doi.org/10.1093/mnras/stac064>