Score Matching and Its Role in Score-based (Diffusion) Generative Models

1 Introduction

Many powerful generative models define densities only up to a normalization constant, $p_\theta(x) = \tilde{p}_\theta(x)/Z(\theta)$, which makes maximum-likelihood learning difficult. *Score matching* circumvents the partition-function obstacle by directly learning the *score*

$$s(x) = \nabla_x \log p(x),$$

the gradient of the log-density, which does not depend on $Z(\theta)$. Modern score-based/diffusion models first corrupt data with noise and then *reverse* that corruption using a neural network trained to approximate scores at different noise levels.

2 Score and Score Matching

Let $s_\theta(x) = \nabla_x \log p_\theta(x)$ be a parametric *score network*. Hyvärinen's score matching learns s_θ by minimizing the Fisher divergence between model and data:

$$\min_\theta \ \mathbb{E}_{p(x)}\left[\|s_\theta(x) - \nabla_x \log p(x)\|_2^2\right]. \tag{1}$$

Using integration by parts under mild regularity conditions, this becomes a tractable objective that requires no access to the data score:

$$\mathcal{J}(\theta) = \mathbb{E}_{p(x)}\left[\tfrac{1}{2}\|s_\theta(x)\|_2^2 + \operatorname{div} s_\theta(x)\right] + \text{const}, \tag{2}$$

where $\operatorname{div} s(x) = \sum_i \partial s_i(x)/\partial x_i$ is the divergence. Intuitively, if the model's score field matches the true data score everywhere, gradient ascent on $\log p$ pushes noisy samples toward the data manifold.

2.1 Denoising Score Matching (DSM)

The divergence term can be costly to estimate; DSM provides a practical surrogate. Corrupt data with Gaussian noise at scale σ: $\tilde{x} = x + \sigma z,\ z \sim \mathcal{N}(0, I)$. Train a noise-conditional network $s_\theta(\tilde{x}, \sigma)$ with

$$\min_\theta \ \mathbb{E}_{x,z}\left[\lambda(\sigma)\left\|s_\theta(x + \sigma z, \sigma) + \tfrac{1}{\sigma} z\right\|_2^2\right]. \tag{3}$$

Because for Gaussian corruption $\nabla_{\tilde{x}} \log q(\tilde{x} \mid x) = -(\tilde{x} - x)/\sigma^2 = -z/\sigma$, the optimal predictor of this conditional score equals the marginal score $\nabla_{\tilde{x}} \log p_\sigma(\tilde{x})$. In practice one uses multiple σ values and a weight $\lambda(\sigma)$ (often proportional to σ^2) to balance scales.

Remark (bounded/discrete data). Images lie in a bounded/discrete support (e.g., pixels in $[0, 1]$ or $\{0, \ldots, 255\}$). Adding Gaussian noise after *dequantization* or a *logit* transform makes the Gaussian corruption assumption better aligned with the data domain and improves training stability.

2.2 Sliced Score Matching (SSM) via Hutchinson's Estimator

In high dimensions, directly computing the divergence term in (2) is expensive because it involves the trace of the Jacobian $\nabla_x s_\theta(x)$. Hutchinson's trick estimates traces with random probe vectors v:

$$\mathrm{tr}(A) = \mathbb{E}_v\left[v^\top A v\right], \quad v \sim \mathcal{N}(0, I) \text{ or Rademacher.}$$

Using this for $A = \nabla_x s_\theta(x)$ yields an equivalent, scalable objective (the *sliced score matching* loss):

$$\mathcal{J}_{\mathrm{SSM}}(\theta) = \mathbb{E}_{p(x)}\mathbb{E}_v\left[\tfrac{1}{2}\|s_\theta(x)\|_2^2 + v^\top \nabla_x s_\theta(x)\, v\right]. \tag{4}$$

Noting that $v^\top \nabla_x s_\theta(x)\, v = \nabla_x\big(v^\top s_\theta(x)\big)^\top v$ is the *directional derivative* of s_θ along v, (4) can be computed with efficient vector–Jacobian products. In practice, one or a few probe vectors per sample already provide a good, unbiased estimate of the divergence term.

3 From Scores to Generative Modeling

Score-based generative modeling defines a forward noising process $\{p_t\}_{t\in[0,1]}$ that gradually turns data into easy-to-sample noise, then *reverses* it using learned scores.

Forward (diffusion) processes. Two widely used continuous-time SDEs are:

$$\text{VE-SDE:} \qquad \mathrm{d}x = g(t)\,\mathrm{d}W_t, \tag{5}$$

$$\text{VP-SDE:} \qquad \mathrm{d}x = -\tfrac{1}{2}\beta(t)\,x\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}W_t, \tag{6}$$

where W_t is standard Brownian motion, and $g(t), \beta(t)$ define the noise schedule. Training uses DSM to learn a time-conditioned score $s_\theta(x,t) \approx \nabla_x \log p_t(x)$.

Reverse-time dynamics for sampling. By Anderson's theorem, the reverse-time SDE is

$$\mathrm{d}x = \big[f(x,t) - g(t)^2 \nabla_x \log p_t(x)\big]\mathrm{d}t + g(t)\,\mathrm{d}\bar{W}_t, \tag{7}$$

where f, g are the drift/diffusion of the forward SDE and $\bar{W}_t$ is a reverse-time Brownian motion. Replacing the unknown score with $s_\theta(x,t)$ yields a sampler that maps noise at $t = 1$ back to data at $t = 0$. A deterministic alternative is the *probability-flow ODE*:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x,t) - \tfrac{1}{2}g(t)^2\, s_\theta(x,t), \tag{8}$$

which can be solved with standard ODE solvers and enables likelihood computation via instantaneous change-of-variables.

4 Connection to DDPM

Discrete-time DDPMs instantiate a VP process with $x_t = \sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon$. If a model predicts noise $\varepsilon_\theta(x_t, t)$, it is equivalent to a score model via

$$\text{VP/DDPM:} \qquad s_\theta(x_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(x_t, t). \tag{9}$$

For VE parameterizations with $x_t = x_0 + \sigma_t z$, we similarly have

$$\text{VE:} \qquad s_\theta(x_t, t) = -\frac{1}{\sigma_t}\,\varepsilon_\theta(x_t, t). \tag{10}$$

Thus "noise prediction" and "score prediction" are two parameterizations of the same underlying vector field.

5 Practical Considerations and Common Pitfalls

- **Noise schedule and weighting.** Choose $\{\sigma\}$ or $\beta(t)$ to cover a wide SNR range; balance scales with $\lambda(\sigma)$ to avoid domination by either extremely large or small noise.
- **Discretization.** Too few reverse steps introduce bias; higher-order integrators or adaptive solvers can reduce error and step count.
- **Stochastic vs. deterministic sampling.** Reverse SDE with predictor–corrector or Langevin refinements often improves fidelity; ODE sampling is faster and supports exact likelihoods.
- **Data support.** For bounded or discrete data (e.g., images in $[0, 1]$), use dequantization or logit transforms before adding Gaussian noise.
- **When not using DSM.** The divergence term in (2) can be estimated efficiently using Hutchinson's trick as in (4).

6 Conclusion

Score matching learns the gradient of the log-density without computing normalization constants. In score-based/diffusion models, a time-conditioned score network trained with denoising objectives provides the force term that reverses a carefully designed noising process, turning random noise into realistic samples. DDPMs are a discrete instantiation of the same principle, differing mainly in parameterization and numerical integration.

References

[1] A. Hyvärinen. Estimation of non-normalizable statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[2] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[3] Y. Song and S. Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *NeurIPS*, 2019.

[4] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. In *NeurIPS*, 2020.

[5] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021.

[6] Y. Song and S. Ermon. Sliced Score Matching: A Scalable Approach to Density and Score Estimation. *arXiv:1905.07088*, 2019.

[7] This report uses AI tools for expression optimization and data search. The core analysis and conclusions remain the responsibility of the authors.