Score-Based Generative Models: Score Matching

Nathan Cantafio

April 16, 2026

Score matching provides a principled route to learning the score function $\nabla_x \log p(x)$ without requiring evaluation of the density itself. This note begins with the ideal regression objective and derives the tractable score matching loss via an integration-by-parts argument. We then examine the limitations of this formulation, including the expensive trace term and the failure of the full-support assumption for real data. These issues motivate denoising score matching, which replaces the original objective with a simple regression problem in which the model learns to remove Gaussian noise.

We conclude by showing how these ideas extend naturally to multi-scale training, annealed Langevin dynamics, and modern diffusion models, where a time-dependent score is learned across noise levels and used to define a reverse-time stochastic process.

1. Introduction

The central problem of generative modelling is sampling from a distribution $p(x)$ given samples alone. A natural approach is to directly estimate the density, either explicitly or up to a normalizing constant. However, in high dimensions this is notoriously difficult: normalizing constants are intractable, likelihoods can be poorly behaved, and models that admit efficient sampling often impose strong structural constraints.

Score matching offers a different perspective. Rather than modelling the density itself, we instead learn its gradient,

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

known as the *score function* (or *Stein score*).. This quantity is sufficient for sampling via Langevin dynamics, and crucially, it does not depend on the normalizing constant of p.

At first glance, this may seem like a minor reparameterization. However, it leads to a fundamentally different learning problem. Instead of likelihood maximization, we obtain an objective that depends only on derivatives of the model and expectations over data. After further refinement, this reduces to a simple denoising regression task.

This note develops score matching from first principles. We begin with the ideal but intractable objective and derive the tractable formulation of [1]. We then examine its limitations and motivate denoising score matching [2]. Finally, we show how these ideas naturally extend to multi-scale training and modern diffusion models.

2. Why learn the score?

It is natural to ask why we would learn the score at all, rather than the density $p(x)$ or an unnormalized density $P(x)$. In principle, if we can evaluate $P(x)$ up to a constant, we can use a local sampler such as Metropolis–Hastings that requires only an unnormalized density to sample from p.

The primary motivation for learning the score instead is a structural one. A scalar density model must satisfy strong constraints: the output must be positive and the function must be globally integrable. These constraints are difficult to enforce for arbitrary neural network architectures, and likelihood-based training can be unstable or computationally expensive.

In contrast, the score is a vector field with no positivity or integrability constraints, making it a more natural target for unconstrained function approximation.

There is also an empirical observation worth noting: gradient-guided samplers such as Langevin dynamics and Hamiltonian Monte Carlo tend to mix well in high-dimensional spaces compared to gradient-free methods like random-walk Metropolis–Hastings, whose acceptance rates degrade rapidly with dimension. Whether this constitutes a principled argument *for* learning scores over densities, or merely a property of certain sampling algorithms, is a subtle distinction. It may be more accurate to say that score-based methods are naturally compatible with the class of samplers that appear to work well in high dimensions.

3. Score Matching

3.1. The problem

We would like to learn the score function

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

given only samples from an unknown density $p(x)$. As discussed previously, the score is sufficient for sampling, so this is a viable route to generative modelling.

Let $s_\theta(x)$ be a neural network approximating the score. The ideal objective would be

$$\mathcal{J}_{\text{ideal}}(\theta) = \frac{1}{2}\,\mathbb{E}_{p(x)}\left[\|s_\theta(x) - \nabla_x \log p(x)\|^2\right]. \tag{1}$$

However, this is not computable: the target $\nabla_x \log p(x)$ is precisely what we do not know. Score matching provides a way around this.

3.2. From the ideal objective to a tractable one

We expand the squared norm in (1) component-wise:

$$\mathcal{J}_{\text{ideal}}(\theta) = \frac{1}{2}\mathbb{E}_p\left[\|s_\theta(x)\|^2 - 2s_\theta(x)^\top \nabla_x \log p(x) + \|\nabla_x \log p(x)\|^2\right].$$

The final term does not depend on θ and can be ignored. We focus on eliminating the cross-term. Writing $\nabla_x \log p(x) = \nabla_x p(x)/p(x)$ and expanding the expectation as an integral:

$$\mathbb{E}_p\Big[s_\theta(x)^\top \nabla_x \log p(x)\Big] = \int p(x)\, s_\theta(x)^\top \frac{\nabla_x p(x)}{p(x)}\, dx = \int s_\theta(x)^\top \nabla_x p(x)\, dx.$$

We now apply integration by parts to each component i separately. Recall that for scalar functions $f, g : \mathbb{R} \to \mathbb{R}$,

$$\int f(t)\, g'(t)\, dt = \big[f(t)g(t)\big]_{-\infty}^{\infty} - \int f'(t)\, g(t)\, dt.$$

Applying this to the i-th component of $s_\theta(x)$ with $g = p$, and treating all other coordinates as fixed:

$$\int s_\theta^{(i)}(x)\, \frac{\partial p(x)}{\partial x_i}\, dx = \Big[s_\theta^{(i)}(x)\, p(x)\Big]_\partial - \int \frac{\partial s_\theta^{(i)}(x)}{\partial x_i}\, p(x)\, dx.$$

The boundary term $[s_\theta^{(i)}(x)\, p(x)]_\partial$ vanishes provided $p(x) \to 0$ sufficiently fast as $\|x\| \to \infty$, which we assume throughout. Summing over all components:

$$\int s_\theta(x)^\top \nabla_x p(x)\, dx = -\int p(x) \sum_{i=1}^{d} \frac{\partial s_\theta^{(i)}(x)}{\partial x_i}\, dx = -\mathbb{E}_p[\nabla_x \cdot s_\theta(x)]\,.$$

Substituting back into the expanded objective and dropping the constant:

$$\mathcal{J}(\theta) = \mathbb{E}_p\left[\operatorname{tr}(\nabla_x s_\theta(x)) + \frac{1}{2}\|s_\theta(x)\|^2\right], \tag{2}$$

which is the *score matching objective* of [1]. This expression depends only on samples from $p(x)$ and the Jacobian of the model, and is therefore computable without knowledge of p.

3.3. Why this works

The key point is that we have constructed an objective that differs from the ideal objective only by a constant in θ. Therefore, the minimiser satisfies

$$s_{\theta^\star}(x) = \nabla_x \log p(x) \quad \text{almost surely.}$$

Conceptually, score matching works by replacing the unknown score with a quantity that can be recovered through integration by parts. The price we pay is the appearance of the divergence term $\nabla_x \cdot s_\theta(x)$.

3.4. The trace problem and the manifold problem

The divergence term in (2) is

$$\operatorname{tr}(\nabla_x s_\theta(x)) = \sum_{i=1}^{d} \frac{\partial s_\theta^{(i)}(x)}{\partial x_i}.$$

Computing this exactly requires d backward passes per data point—prohibitively expensive in high dimensions such as images. One can instead use Hutchinson's estimator,

$$\operatorname{tr}(A) = \mathbb{E}_v[v^\top A v], \quad v \sim \mathcal{N}(0, I),$$

which reduces the cost to a small number of Jacobian-vector products. This approach was explored in the sliced score matching objective of [3], which replaces the full trace with random projections and is practical for moderately-sized problems. However, variance in the estimator can still limit performance at scale.

Beyond computational cost, there is a more fundamental obstruction. The integration-by-parts argument used to derive (2) assumes that $p(x)$ has full support over $\mathbb{R}^d$ (in particular, that the boundary term vanishes because p decays smoothly). Real data distributions are believed to concentrate near low-dimensional manifolds embedded in $\mathbb{R}^d$. On such manifolds, the score is undefined or inconsistently estimated off the support, providing no useful learning signal in the ambient space.

Both the expensive trace and the manifold structure motivate denoising score matching.

4. Denoising Score Matching

4.1. Adding noise

To address the manifold issue, we perturb the data:

$$\tilde{x} = x + \sigma z, \quad z \sim \mathcal{N}(0, I).$$

This defines a new density

$$\tilde{p}_\sigma(\tilde{x}) = \int p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2 I)\, dx,$$

which has full support over $\mathbb{R}^d$, satisfies the boundary conditions required for integration by parts, and provides learning signal everywhere in the ambient space.

We can now apply score matching to $\tilde{p}_\sigma$ instead of p.

4.2. The key identity

We wish to compute $\nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x})$. Begin by differentiating the definition of $\tilde{p}_\sigma$ directly:

$$\nabla_{\tilde{x}} \tilde{p}_\sigma(\tilde{x}) = \nabla_{\tilde{x}} \int p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2 I)\, dx = \int p(x)\, \nabla_{\tilde{x}} \mathcal{N}(\tilde{x}; x, \sigma^2 I)\, dx,$$

where we have passed the derivative under the integral (justified by dominated convergence under mild regularity). Using $\nabla_{\tilde{x}} f = f \cdot \nabla_{\tilde{x}} \log f$ for any positive f, we write:

$$\nabla_{\tilde{x}} \mathcal{N}(\tilde{x}; x, \sigma^2 I) = \mathcal{N}(\tilde{x}; x, \sigma^2 I) \cdot \nabla_{\tilde{x}} \log \mathcal{N}(\tilde{x}; x, \sigma^2 I).$$

For the Gaussian, the log-density is $-\frac{1}{2\sigma^2}\|\tilde{x} - x\|^2 + \text{const}$, so:

$$\nabla_{\tilde{x}} \log \mathcal{N}(\tilde{x}; x, \sigma^2 I) = -\frac{\tilde{x} - x}{\sigma^2}.$$

Substituting:

$$\nabla_{\tilde{x}} \tilde{p}_\sigma(\tilde{x}) = \int p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2 I) \left(-\frac{\tilde{x} - x}{\sigma^2} \right) dx.$$

Dividing both sides by $\tilde{p}_\sigma(\tilde{x})$ to obtain the score:

$$\nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x}) = \frac{\nabla_{\tilde{x}} \tilde{p}_\sigma(\tilde{x})}{\tilde{p}_\sigma(\tilde{x})} = \int \frac{p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2 I)}{\tilde{p}_\sigma(\tilde{x})} \left(-\frac{\tilde{x} - x}{\sigma^2} \right) dx.$$

The factor $p(x)\, \mathcal{N}(\tilde{x}; x, \sigma^2 I) / \tilde{p}_\sigma(\tilde{x})$ is precisely the posterior density $p(x \mid \tilde{x})$ by Bayes' theorem, since:

$$p(x \mid \tilde{x}) = \frac{p(\tilde{x} \mid x)\, p(x)}{p(\tilde{x})} = \frac{\mathcal{N}(\tilde{x}; x, \sigma^2 I)\, p(x)}{\tilde{p}_\sigma(\tilde{x})}.$$

We therefore obtain:

$$\nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x}) = \mathbb{E}_{p(x \mid \tilde{x})} \left[-\frac{\tilde{x} - x}{\sigma^2} \right]. \tag{3}$$

This is a remarkable identity: *the score of the noisy marginal is the posterior expectation of the denoising direction*. The local gradient of the log-density at a noisy point $\tilde{x}$ is a weighted average of the directions pointing back toward each plausible clean data point x, with weights given by the posterior $p(x \mid \tilde{x})$. The score thus encodes where the data is, averaged over posterior uncertainty about which clean point generated $\tilde{x}$.

4.3. From score matching to denoising

Consider applying the ideal score matching objective to $\tilde{p}_\sigma$:

$$\mathcal{J}_{\text{ideal}}(\theta; \sigma) = \frac{1}{2}\, \mathbb{E}_{\tilde{p}_\sigma(\tilde{x})} \left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x})\|^2 \right]. \tag{4}$$

We claim this is equal, up to a constant in θ, to the joint objective

$$\mathcal{J}_{\text{DSM}}(\theta; \sigma) = \frac{1}{2}\, \mathbb{E}_{p(x)}\, \mathbb{E}_{p(\tilde{x} \mid x)} \left[\left\| s_\theta(\tilde{x}) + \frac{\tilde{x} - x}{\sigma^2} \right\|^2 \right]. \tag{5}$$

To see this, expand (5) by writing out the squared norm:

$$\mathcal{J}_{\text{DSM}}(\theta; \sigma) = \frac{1}{2}\, \mathbb{E}\left[\|s_\theta(\tilde{x})\|^2 \right] + \mathbb{E}\left[s_\theta(\tilde{x})^\top \frac{\tilde{x} - x}{\sigma^2} \right] + \underbrace{\frac{1}{2}\, \mathbb{E}\left[\left\| \frac{\tilde{x} - x}{\sigma^2} \right\|^2 \right]}_{\text{const in } \theta}.$$

Applying the tower property to the cross-term, conditioning on $\tilde{x}$:

$$\mathbb{E}\left[s_\theta(\tilde{x})^\top \frac{\tilde{x}-x}{\sigma^2}\right] = \mathbb{E}_{\tilde{p}_\sigma(\tilde{x})}\left[s_\theta(\tilde{x})^\top \mathbb{E}_{p(x|\tilde{x})}\left[\frac{\tilde{x}-x}{\sigma^2}\right]\right] = -\mathbb{E}_{\tilde{p}_\sigma}\left[s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x})\right],$$

where the last equality uses (3). Now expand (4) similarly:

$$\mathcal{J}_{\text{ideal}}(\theta;\sigma) = \frac{1}{2}\,\mathbb{E}_{\tilde{p}_\sigma}\left[\|s_\theta(\tilde{x})\|^2\right] - \mathbb{E}_{\tilde{p}_\sigma}\left[s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x})\right] + \underbrace{\frac{1}{2}\,\mathbb{E}_{\tilde{p}_\sigma}\left[\|\nabla_{\tilde{x}} \log \tilde{p}_\sigma(\tilde{x})\|^2\right]}_{\text{const in } \theta}.$$

The first two terms are identical to those in $\mathcal{J}_{\text{DSM}}$, so the two objectives differ only by constants in θ, and share the same minimiser. In practice, (5) is evaluated by sampling $x \sim p(x)$ from the data and $\tilde{x} = x + \sigma z$ with $z \sim \mathcal{N}(0, I)$, making it trivially computable with no trace term.

4.4. Interpretation

Writing $\tilde{x} = x + \sigma z$, the target in (5) becomes

$$-\frac{\tilde{x}-x}{\sigma^2} = -\frac{z}{\sigma}.$$

Thus, the network learns to map a noisy input $\tilde{x}$ to the scaled direction pointing back toward the clean data x. In other words: *learning the score is equivalent to learning how to denoise.*

This perspective is both computationally convenient and conceptually powerful. It eliminates the trace term entirely and replaces it with a simple regression problem. It also reveals a direct connection between score estimation and minimum mean-squared-error denoising.

4.5. Why this is better

Denoising score matching resolves both major issues with the original formulation:

- It avoids the expensive trace computation: the objective (5) requires only forward passes through s_θ.
- It ensures the perturbed density has full support, providing learning signal everywhere in the ambient space.

Moreover, it introduces a natural notion of *scale*: different noise levels σ correspond to different levels of detail in the data, from coarse global structure at large σ to fine local detail at small σ. This observation becomes central in modern diffusion models.

5. From DSM to Annealed Langevin Dynamics

Song and Ermon [4] exploit DSM via *annealed Langevin dynamics*. Let $\sigma_1 > \sigma_2 > \cdots > \sigma_L$ be a sequence of noise levels. Sampling proceeds by running Langevin dynamics at each level in sequence:

$$x \leftarrow x + \frac{\epsilon}{2} s_\theta(x, \sigma_i) + \sqrt{\epsilon}\, z.$$

The intuition is that at large σ, the distribution is smooth and easy to explore globally, while at small σ, the dynamics refine the sample locally toward the data manifold. Rather than training separate models for each σ, a single *noise-conditional score network* $s_\theta(x, \sigma)$ is trained using a weighted objective:

$$\mathcal{L}(\theta) = \sum_{i=1}^{L} \lambda(\sigma_i)\, \mathcal{J}_{\text{DSM}}(\theta;\sigma_i). \tag{6}$$

Empirically, $\|s_\theta(x,\sigma)\| \propto 1/\sigma$, so choosing $\lambda(\sigma) = \sigma^2$ approximately balances the magnitude of each term. This construction forms the basis of early score-based generative models.

6. Time-dependent scores and diffusion models

Modern diffusion models can be understood as a continuous-time generalization of the multi-scale construction above. Instead of a discrete sequence of noise levels, we consider a continuous process $x(t)$ obtained by gradually corrupting data with Gaussian noise over a time interval $[0, T]$. At each time t, the marginal $p_t(x)$ corresponds to a particular noise scale. The goal is to learn the time-dependent score

$$s_\theta(x, t) \approx \nabla_x \log p_t(x).$$

In practice, this is done using a denoising objective analogous to (5). For a noise schedule $(\alpha(t), \sigma(t))$ such that $x_t = \alpha(t)\, x_0 + \sigma(t)\, \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, one can equivalently train the network to predict the noise:

$$\mathbb{E}_{t, x_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right].$$

This is the objective used in DDPM [5]. Predicting the score is related to predicting the noise by $s_\theta(x_t, t) = -\epsilon_\theta(x_t, t)/\sigma(t)$. The learned model defines a reverse-time stochastic differential equation that gradually transforms noise into data, as made precise in the SDE framework of [6].

From this perspective, denoising score matching is not just a trick for avoiding the trace term. It is the fundamental learning principle underlying diffusion models: learning to reverse a noising process by estimating the direction pointing back toward the data.

7. Discussion and Conclusion

We have traced a path from the basic idea of learning the score function to the training objective at the heart of modern diffusion models. It is worth pausing to appreciate what has been established at each step, and what has been left open.

The core insight of score matching [1] is that the score can be learned without ever evaluating $p(x)$, using only an integration-by-parts identity. The resulting objective is computable from samples. Its practical limitations (the expensive trace term and the manifold problem) are not superficial implementation details but reflect genuine structural constraints of the ambient-space formulation. Hutchinson-style estimators and sliced score matching partially address the former; denoising score matching [2] resolves both simultaneously.

The equivalence between score estimation and denoising is perhaps the deepest result in these notes. It reveals that learning to denoise is not merely a convenient approximation but an exact reformulation of the score matching problem for perturbed distributions. This also places diffusion models in direct correspondence with classical MMSE denoising theory.

Several important points lie beyond the scope of these notes. First, there is a gap between the score of the noisy distribution $\tilde{p}_\sigma$ and the score of the true data distribution p: at small σ these are approximately equal, but the approximation breaks down as σ grows, and the effect of this gap on sample quality is non-trivial. Second, the choice of noise schedule has a significant practical impact on model performance and is an active area of research.

What these notes have shown is that the score is not an arbitrary alternative representation of p. It is, in a precise sense, the right primitive for distribution learning in high dimensions: it encodes the local geometry of the distribution, is unconstrained as a function, is directly learnable from data, and is sufficient for sampling. The diffusion model framework is, at its core, a systematic way of exploiting this fact across a continuum of scales.

References

[1] A. Hyvärinen. "Estimation of non-normalized statistical models by score matching". *Journal of Machine Learning Research* 6 (2005), pp. 695–709.

[2] P. Vincent. "A connection between score matching and denoising autoencoders". *Neural Computation* 23.7 (2011), pp. 1661–1674.

[3] Y. Song et al. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation". *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*. UAI'20. PMLR, 2020, pp. 574–584.

[4] Y. Song and S. Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. NeurIPS'19. 2019, pp. 11918–11930.

[5] J. Ho, A. Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models". *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NeurIPS'20. 2020, pp. 6840–6851.

[6] Y. Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.