Denoising Score Matching: The Denoising Principle Behind Scores

Jiahui Zhu

Ph.D Candidate at EECS

February 10, 2026

Outline

1. **Why Score Matching?**
2. **Why Denoising Score Matching (DSM)?**
3. **What techniques can further improve DSM?**
4. **What can we do when clean data are unavailable?**

Why Score Matching? EBM MLE is Intractable

Energy-Based Model (EBM):

$$p_\theta(\boldsymbol{x}) = \frac{\exp(-E_\theta(\boldsymbol{x}))}{Z_\theta}, \qquad Z_\theta = \int \exp(-E_\theta(\boldsymbol{x}))\, d\boldsymbol{x}.$$

Maximum Likelihood (negative log-likelihood):

$$\mathcal{L}_{\text{MLE}}(\theta) = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\big[\log p_\theta(\boldsymbol{x})\big] = \mathbb{E}_{p_{\text{data}}}\big[E_\theta(\boldsymbol{x})\big] + \log Z_\theta.$$

Key issue: $\log Z_\theta$ involves a high-dimensional integral.
Its gradient requires sampling from p_θ:

$$\nabla_\theta \log Z_\theta = \mathbb{E}_{\boldsymbol{x}\sim p_\theta}\big[\nabla_\theta E_\theta(\boldsymbol{x})\big] \quad \Rightarrow \quad \text{intractable / expensive in high-dim.}$$

Idea: learn $\nabla_x \log p_\theta(\boldsymbol{x})$ instead of $\log p_\theta(\boldsymbol{x})$, because $\nabla_x \log Z_\theta = 0$.

Score and the Original Score Matching Objective

Score function (data vs. model):

$$\boldsymbol{s}_{\text{data}}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x}), \qquad \boldsymbol{s}_\theta(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}).$$

For an EBM: $p_\theta(\boldsymbol{x}) \propto \exp(-E_\theta(\boldsymbol{x})) \Rightarrow$

$$\boldsymbol{s}_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x}) \qquad (\text{since } \nabla_{\boldsymbol{x}} \log Z_\theta = 0).$$

Original score matching (conceptual):

$$\min_\theta \ \frac{1}{2}\, \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \left[\|\boldsymbol{s}_\theta(\boldsymbol{x}) - \boldsymbol{s}_{\text{data}}(\boldsymbol{x})\|^2 \right].$$

Challenge: $\boldsymbol{s}_{\text{data}}(\boldsymbol{x})$ is unknown (we only have samples).

Hyvärinen Score Matching: Computable Form (but 2nd-order)

Let $s_\theta(\boldsymbol{x})$ be any score network (ideally $s_\theta = \nabla \log p_\theta$). Hyvärinen showed the objective is equivalent to:

$$\min_\theta \ \mathbb{E}_{p_{\text{data}}}\Big[\frac{1}{2}\|s_\theta(\boldsymbol{x})\|^2 + \text{div}_{\boldsymbol{x}}\ s_\theta(\boldsymbol{x})\Big],$$

where $\text{div}_{\boldsymbol{x}}\ s_\theta(\boldsymbol{x}) = \text{Tr}\big(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})\big)$.

If $s_\theta(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x})$, then

$$\text{div}_{\boldsymbol{x}}\ s_\theta(\boldsymbol{x}) = -\text{Tr}\big(\nabla^2_{\boldsymbol{x}} E_\theta(\boldsymbol{x})\big),$$

and the loss becomes :

$$\mathbb{E}_{p_{\text{data}}}\Big[\frac{1}{2}\|\nabla_{\boldsymbol{x}} E_\theta(\boldsymbol{x})\|^2 + \text{Tr}\big(\nabla^2_{\boldsymbol{x}} E_\theta(\boldsymbol{x})\big)\Big].$$

Bottleneck: computing $\text{Tr}(\nabla^2_{\boldsymbol{x}} E_\theta)$ (Hessian trace) is costly in high dimension.

Sliced Score Matching and Hutchinson's Estimator

Goal: replace the trace in score matching by averaging directional derivatives along random "slices".

Let $\boldsymbol{v} \in \mathbb{R}^D$ be an *isotropic* random vector (e.g., Rademacher or standard Gaussian) with $\mathbb{E}[\boldsymbol{v}] = 0$ and $\mathbb{E}[\boldsymbol{v}\boldsymbol{v}^\top] = I$. By Hutchinson's identity,

$$\mathrm{Tr}(A) = \mathbb{E}_{\boldsymbol{v}}\big[\boldsymbol{v}^\top A\, \boldsymbol{v}\big], \qquad \mathbb{E}_{\boldsymbol{v}}\big[(\boldsymbol{v}^\top s_\theta(\boldsymbol{x}))^2\big] = \|s_\theta(\boldsymbol{x})\|^2.$$

Thus we obtain the **exact** sliced score matching objective:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}},\,\boldsymbol{v}}\Big[\boldsymbol{v}^\top\big(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})\big)\boldsymbol{v} + \frac{1}{2}\big(\boldsymbol{v}^\top s_\theta(\boldsymbol{x})\big)^2\Big].$$

Implementation note:

$$\boldsymbol{v}^\top(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}))\boldsymbol{v} = \nabla_{\boldsymbol{x}}\big(s_\theta(\boldsymbol{x})^\top \boldsymbol{v}\big)^\top \boldsymbol{v},$$

so the objective can be evaluated efficiently with autodiff using JVP/VJP (no explicit Jacobian/Hessian). Averaging over K probes gives an unbiased estimator with variance $O(1/K)$.

Sliced SM: Pros and Cons (Why We Still Need DSM)

Pros:

- Avoid explicit Hessian/Jacobian construction; scalable to high dimension.
- Uses only vector-Jacobian products

Cons (practical pain points):

- **Estimator variance:** Hutchinson trace estimate can be noisy $\Rightarrow$ unstable training.
- **Clean-data score issues:** if p_{data} concentrates near a low-dimensional manifold, the true score $\nabla \log p_{\text{data}}(\boldsymbol{x})$ can be ill-behaved.
- **Sampling/mixing:** learning only the *clean* score does not directly give a multi-noise-level field; Langevin sampling can mix poorly for complex multimodal data.

Remedy: add noise $\Rightarrow$ learn the score of a *smoothed* density p_σ via **Denoising Score Matching (DSM)**.

Bridge: From DSM to the Denoising View (Why Vincent 2011)

We just motivated **DSM** by adding noise and learning the score of a smoothed density p_σ.

Key shift: learn a *denoiser* (regression) instead of computing divergence/Hessian:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \qquad \text{learn } D_\theta(\tilde{\boldsymbol{x}}) \approx \boldsymbol{x}.$$

Why this is powerful:

- Avoids expensive second-order terms (Hessian/divergence) in original SM / sliced SM.
- Noise makes p_σ smoother with full support $\Rightarrow$ score field is well-behaved.

Next: Vincent (2011) explains the core connection:

$$\text{(denoising residual)} \iff \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}).$$

This deepens why DSM works and links it to classical denoising autoencoders.

Vincent (2011): Denoising Autoencoder (DAE) Objective

Noisy observation:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \qquad p_\sigma(\tilde{\boldsymbol{x}}) = \int p(\boldsymbol{x})\,\mathcal{N}(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \sigma^2 I)\,d\boldsymbol{x}.$$

DAE / denoising regression loss: learn a denoiser $D_\theta(\cdot)$ by

$$\min_\theta \; \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}|\boldsymbol{x}}\Big[\,\|D_\theta(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2\,\Big].$$

What this buys: a clean regression objective whose optimum is the Bayes denoiser

$$D^*(\tilde{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}].$$

Why we need this optimum?

Why the Bayes Denoiser is the Conditional Mean

Goal: for each observed $\tilde{\boldsymbol{x}}$, choose $D(\tilde{\boldsymbol{x}})$ to minimize the MSE

$$\min_{D} \ \mathbb{E}\big[\, \|D(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2 \,\big].$$

Key step: optimize pointwise for each fixed $\tilde{\boldsymbol{x}}$:

$$D^*(\tilde{\boldsymbol{x}}) = \arg\min_{d \in \mathbb{R}^d} \ \mathbb{E}\big[\, \|d - \boldsymbol{x}\|^2 \mid \tilde{\boldsymbol{x}} \big].$$

Expand the conditional risk:

$$\mathbb{E}\big[\, \|d - \boldsymbol{x}\|^2 \mid \tilde{\boldsymbol{x}} \big] = \mathbb{E}[\|\boldsymbol{x}\|^2 \mid \tilde{\boldsymbol{x}}] - 2d^\top \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] + \|d\|^2 .$$

Differentiate w.r.t. d and set to zero:

$$\nabla_d(\cdot) = -2\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] + 2d = 0 \quad \Rightarrow \quad D^*(\tilde{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}].$$

Takeaway: the denoising MSE objective learns the posterior mean.

Deriving Tweedie's Formula

Recall the noisy marginal:

$$p_\sigma(\tilde{\boldsymbol{x}}) = \int p(\boldsymbol{x})\,\mathcal{N}(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \sigma^2 I)\, d\boldsymbol{x}.$$

Differentiate w.r.t. $\tilde{\boldsymbol{x}}$:

$$\nabla_{\tilde{\boldsymbol{x}}} p_\sigma(\tilde{\boldsymbol{x}}) = \int p(\boldsymbol{x})\,\nabla_{\tilde{\boldsymbol{x}}}\mathcal{N}(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \sigma^2 I)\, d\boldsymbol{x}.$$

For Gaussian,

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

Plug in:

$$\nabla_{\tilde{\boldsymbol{x}}} p_\sigma(\tilde{\boldsymbol{x}}) = \int p(\boldsymbol{x})\,\mathcal{N}(\tilde{\boldsymbol{x}}; \boldsymbol{x}, \sigma^2 I)\Big(\frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{\sigma^2}\Big)\, d\boldsymbol{x}.$$

Divide by $p_\sigma(\tilde{\boldsymbol{x}})$:

$$\nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}) = \frac{1}{\sigma^2}\Big(\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] - \tilde{\boldsymbol{x}}\Big).$$

Rearrange (Tweedie):

$$\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] = \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}).$$

Why DSM Works: From Kernel Score to Marginal Score

DSM trains a score network by matching the **corruption kernel score**:

$$\min_{\theta} \ \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{x},\sigma^2 I)}\left[\|\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}},\sigma)-\nabla_{\tilde{\boldsymbol{x}}}\log p(\tilde{\boldsymbol{x}}\mid\boldsymbol{x})\|^2\right].$$

For Gaussian corruption,

$$\nabla_{\tilde{\boldsymbol{x}}}\log p(\tilde{\boldsymbol{x}}\mid\boldsymbol{x})=-\frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma^2}.$$

Key identity: the optimizer of an MSE regression is the conditional mean:

$$\boldsymbol{s}^*_\theta(\tilde{\boldsymbol{x}},\sigma)=\mathbb{E}\big[\nabla_{\tilde{\boldsymbol{x}}}\log p(\tilde{\boldsymbol{x}}\mid\boldsymbol{x})\mid\tilde{\boldsymbol{x}}\big].$$

Compute the RHS using the Gaussian form:

$$\boldsymbol{s}^*_\theta(\tilde{\boldsymbol{x}},\sigma)=\mathbb{E}\big[\nabla_{\tilde{\boldsymbol{x}}}\log p(\tilde{\boldsymbol{x}}\mid\boldsymbol{x})\mid\tilde{\boldsymbol{x}}\big]=\mathbb{E}\Big[-\frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma^2}\,\Big|\,\tilde{\boldsymbol{x}}\Big]=\frac{1}{\sigma^2}\Big(\mathbb{E}[\boldsymbol{x}\mid\tilde{\boldsymbol{x}}]-\tilde{\boldsymbol{x}}\Big)=\nabla_{\tilde{\boldsymbol{x}}}\log p_\sigma(\tilde{\boldsymbol{x}}),$$

where the last equality is exactly Tweedie's formula.

Takeaway: DSM learns the *smoothed marginal score* $\nabla_{\tilde{\boldsymbol{x}}}\log p_\sigma(\tilde{\boldsymbol{x}})$ by regressing to a *known* target $-(\tilde{\boldsymbol{x}}-\boldsymbol{x})/\sigma^2$.

From DAE to DSM: Residual-to-Score and the DSM Training Loss

Bayes denoiser: minimizing $\mathbb{E}\,\|D_\theta(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2$ yields

$$D^*(\tilde{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}].$$

Tweedie:

$$\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] \;=\; \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}) \quad \Rightarrow \quad \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}) = \frac{\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] - \tilde{\boldsymbol{x}}}{\sigma^2}.$$

So we can parameterize the denoiser via a score network:

$$D_\theta(\tilde{\boldsymbol{x}}) := \tilde{\boldsymbol{x}} + \sigma^2\, \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}, \sigma) \quad \Leftrightarrow \quad \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}, \sigma) = \frac{D_\theta(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}}{\sigma^2}.$$

Plug into DAE loss:

$$\|D_\theta(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2 = \left\|\tilde{\boldsymbol{x}} + \sigma^2 \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}, \sigma) - \boldsymbol{x}\right\|^2 = \sigma^4 \left\|\boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}, \sigma) - \frac{\boldsymbol{x} - \tilde{\boldsymbol{x}}}{\sigma^2}\right\|^2.$$

With $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$,

$$\frac{1}{2\sigma^4}\,\mathbb{E}\big[\,\|D_\theta(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2\,\big] = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}}\Big[\,\Big\|\boldsymbol{s}_\theta(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \sigma) + \frac{\boldsymbol{\epsilon}}{\sigma}\Big\|^2\,\Big],$$

Why We Need *High-Order* Tweedie?

We already learned **first-order** score via DSM:

$$\boldsymbol{s}_1(\tilde{\boldsymbol{x}};\sigma) := \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}), \qquad \tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(0, I).$$

But first-order Tweedie only gives a point estimate (posterior mean):

$$\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] \ = \ \tilde{\boldsymbol{x}} + \sigma^2 \boldsymbol{s}_1(\tilde{\boldsymbol{x}};\sigma).$$

Why low-order is not enough:

- **Uncertainty / multi-modality:** many $\boldsymbol{x}$ can explain one $\tilde{\boldsymbol{x}}$; the mean alone hides ambiguity.
- **Curvature matters:** local geometry of $\log p_\sigma$ needs $\nabla^2 \log p_\sigma$ (Hessian).
- **Speed:** differentiating a learned $\boldsymbol{s}_1$ to get Hessians is expensive and amplifies errors.

Goal of Meng et al. (2021): directly learn $\boldsymbol{s}_2(\tilde{\boldsymbol{x}};\sigma) = \nabla^2_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}})$ (and higher), instead of autodiff on $\boldsymbol{s}_1$.

High-Order Tweedie: Mean & Covariance from Scores

Define $\boldsymbol{s}_1(\tilde{\boldsymbol{x}}; \sigma) = \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}})$ and

$$\boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma) := \nabla^2_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}) \quad \text{(a Hessian matrix)}.$$

(1) First-order Tweedie (mean):

$$\mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] \;=\; \tilde{\boldsymbol{x}} + \sigma^2 \boldsymbol{s}_1(\tilde{\boldsymbol{x}}; \sigma).$$

(2) Second-order Tweedie (covariance):

$$\mathrm{Cov}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] \;=\; \sigma^2 I \;+\; \sigma^4 \, \boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma).$$

Interpretation:

- $\boldsymbol{s}_1$: tells you the *denoising direction* (move toward high density).
- $\boldsymbol{s}_2$: tells you the *uncertainty/curvature* (how sharp/flat the posterior is).

Takeaway: high-order Tweedie turns $\boldsymbol{s}_2$ into a *statistical quantity* (posterior covariance), which is something DSM alone cannot provide.

Meng et al. (2021): Key Mechanism (Moments $\leftrightarrow$ High-Order Scores)

High-order scores:

$$\boldsymbol{s}_k(\tilde{\boldsymbol{x}};\sigma) := \nabla_{\tilde{\boldsymbol{x}}}^{k} \log p_\sigma(\tilde{\boldsymbol{x}}), \quad k \geq 1,$$

(tensor-valued for $k \geq 3$).

Core claim (generalized Tweedie idea): higher-order *posterior moments* of $\boldsymbol{x} \mid \tilde{\boldsymbol{x}}$ can be written as polynomials of $\tilde{\boldsymbol{x}}$ and $\boldsymbol{s}_1(\tilde{\boldsymbol{x}};\sigma), \ldots, \boldsymbol{s}_k(\tilde{\boldsymbol{x}};\sigma)$.

Example (2nd moment identity): there exists a polynomial f_2 such that

$$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \mid \tilde{\boldsymbol{x}}] = f_2\big(\tilde{\boldsymbol{x}}, \boldsymbol{s}_1(\tilde{\boldsymbol{x}};\sigma), \boldsymbol{s}_2(\tilde{\boldsymbol{x}};\sigma)\big).$$

How this becomes learnable: we can sample pairs $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ via $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}$, so targets like $\boldsymbol{x}^{\otimes k}$ are *observable* from data.

Concrete Training Objective for s_2 (Second-Order DSM)

Assume we already have a trained first-order score $\boldsymbol{s}_1(\tilde{\boldsymbol{x}}; \sigma)$ from DSM. We parameterize a **second-order score model**:

$$\boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma; \theta) \approx \nabla^2_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}).$$

Polynomial form used for learning (one explicit instantiation):

$$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \mid \tilde{\boldsymbol{x}}] = \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top + \sigma^2 \tilde{\boldsymbol{x}}\boldsymbol{s}_1^\top + \sigma^2 \boldsymbol{s}_1 \tilde{\boldsymbol{x}}^\top + \sigma^4 \boldsymbol{s}_2 + \sigma^4 \boldsymbol{s}_1 \boldsymbol{s}_1^\top + \sigma^2 I,$$

where $\boldsymbol{s}_1 = \boldsymbol{s}_1(\tilde{\boldsymbol{x}}; \sigma)$ and $\boldsymbol{s}_2 = \boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma)$.

Least-squares objective (supervised by x itself):

$$\min_\theta \ \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}),\, \epsilon\sim\mathcal{N}(0,I)} \left\| \boldsymbol{x}\boldsymbol{x}^\top - \left(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top + \sigma^2 \tilde{\boldsymbol{x}}\boldsymbol{s}_1^\top + \sigma^2 \boldsymbol{s}_1 \tilde{\boldsymbol{x}}^\top + \sigma^4 \boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma; \theta) + \sigma^4 \boldsymbol{s}_1 \boldsymbol{s}_1^\top + \sigma^2 I \right) \right\|_F^2.$$

Practical note: in high-d, $\boldsymbol{s}_2$ is often parameterized as diagonal / low-rank for scalability.

What High-Order Tweedie Buys for DSM (and Sampling)

Pipeline view:

- **DSM** $\Rightarrow$ learn $\boldsymbol{s}_1(\tilde{\boldsymbol{x}}; \sigma)$ (denoising direction).
- **High-order DSM** $\Rightarrow$ learn $\boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma)$ (curvature/uncertainty).
- (Optionally) extend to $\boldsymbol{s}_3, \boldsymbol{s}_4, \ldots$ using higher moments $\boldsymbol{x}^{\otimes k}$.

Benefits:

- **Uncertainty quantification:**

$$\mathrm{Cov}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] = \sigma^2 \boldsymbol{I} + \sigma^4 \boldsymbol{s}_2(\tilde{\boldsymbol{x}}; \sigma)$$

  gives pixel-/dimension-wise confidence, not just a mean estimate.
- **Faster / better mixing sampling:** curvature enables higher-order Langevin discretizations (e.g., Ozaki-style local linearization), improving mixing vs. plain Langevin.
- **Avoid autodiff pitfalls:** direct learning of $\boldsymbol{s}_2$ can be far cheaper and more accurate than differentiating a learned $\boldsymbol{s}_1$.

Takeaway: high-order Tweedie turns $\boldsymbol{s}_2$ into *usable statistics* (covariance) and *better samplers*.

SURE Motivation: Denoising Risk *without* Clean Data

Setting:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \qquad \boldsymbol{x} \in \mathbb{R}^d.$$

A denoiser is a mapping

$$D : \mathbb{R}^d \to \mathbb{R}^d, \qquad \hat{\boldsymbol{x}} = D(\tilde{\boldsymbol{x}}).$$

Quality measure (conditional MSE):

$$R(D; \boldsymbol{x}) \;:=\; \mathbb{E}_{\tilde{\boldsymbol{x}}|\boldsymbol{x}}\big[\, \|D(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2 \mid \boldsymbol{x}\big].$$

Problem: $R(D; \boldsymbol{x})$ depends on unknown clean $\boldsymbol{x}$ $\Rightarrow$ cannot be computed from noisy data alone.

Goal: construct an *observable* surrogate using only $\tilde{\boldsymbol{x}}$ that is unbiased for the MSE.

SURE Formula and Unbiasedness

Stein's Unbiased Risk Estimator (SURE):

$$\mathrm{SURE}(D;\tilde{\boldsymbol{x}}) = \|D(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}\|^2 + 2\sigma^2 \,\mathrm{div}_{\tilde{\boldsymbol{x}}}\, D(\tilde{\boldsymbol{x}}) - d\sigma^2,$$

where $\mathrm{div}_{\tilde{\boldsymbol{x}}}\, D(\tilde{\boldsymbol{x}}) = \mathrm{Tr}\big(\nabla_{\tilde{\boldsymbol{x}}} D(\tilde{\boldsymbol{x}})\big)$.

Unbiasedness (Stein's identity): for any fixed (unknown) $\boldsymbol{x}$,

$$\mathbb{E}_{\tilde{\boldsymbol{x}}|\boldsymbol{x}}\big[\mathrm{SURE}(D;\tilde{\boldsymbol{x}}) \mid \boldsymbol{x}\big] \;=\; R(D;\boldsymbol{x}).$$

Averaging over $\boldsymbol{x} \sim p_{\text{data}}$ gives an unbiased estimate of the *global* MSE risk.

Intuition:

- $\|D(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{x}}\|^2$: how much the denoiser changes the noisy input (alone underestimates error).
- $2\sigma^2\,\mathrm{div}_{\tilde{\boldsymbol{x}}}\, D(\tilde{\boldsymbol{x}})$: correction for noise variance (sensitivity of D).

Link to Bayes Optimality and Tweedie's Formula

Taking expectation over $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}$,

$$\mathbb{E}_{\tilde{\boldsymbol{x}}|\boldsymbol{x}}\big[\mathrm{SURE}(D;\tilde{\boldsymbol{x}})\big] = \mathbb{E}_{\tilde{\boldsymbol{x}}|\boldsymbol{x}}\big[\,\|D(\tilde{\boldsymbol{x}}) - \boldsymbol{x}\|^2\,\big].$$

Pointwise (in $\tilde{x}$) Bayes optimal denoiser:

$$D^*(\tilde{\boldsymbol{x}}) = \arg\min_z \mathbb{E}\big[\,\|z - \boldsymbol{x}\|^2 \mid \tilde{\boldsymbol{x}}\big] \;=\; \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}].$$

Let $p_\sigma(\tilde{\boldsymbol{x}}) = (p_{\text{data}} * \mathcal{N}(0, \sigma^2 I))(\tilde{\boldsymbol{x}})$ be the noisy marginal. By Tweedie's formula:

$$D^*(\tilde{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{x} \mid \tilde{\boldsymbol{x}}] \;=\; \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}).$$

Takeaway: SURE-optimal denoising $\Rightarrow$ Bayes denoiser $\Rightarrow$ denoising residual encodes the noisy score.

Relationship of SURE and Score Matching (Why DSM is Denoising)

Motivated by Tweedie, parameterize the denoiser via a score field:

$$D(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}} + \sigma^2 s_\phi(\tilde{\boldsymbol{x}}; \sigma), \qquad s_\phi(\tilde{\boldsymbol{x}}; \sigma) \approx \nabla_{\tilde{\boldsymbol{x}}} \log p_\sigma(\tilde{\boldsymbol{x}}).$$

Plugging into SURE and simplifying yields (up to constants depending only on σ):

$$\frac{1}{2\sigma^4} \mathrm{SURE}(D; \tilde{\boldsymbol{x}}) = \mathrm{Tr}\big(\nabla_{\tilde{\boldsymbol{x}}} s_\phi(\tilde{\boldsymbol{x}}; \sigma)\big) + \frac{1}{2} \left\| s_\phi(\tilde{\boldsymbol{x}}; \sigma) \right\|^2 + \mathrm{const}(\sigma).$$

Taking expectation over $\tilde{\boldsymbol{x}} \sim p_\sigma$, minimizing expected SURE is equivalent (up to an additive constant) to minimizing Hyvärinen's score matching objective under the noisy marginal p_σ:

$$\min_\phi \ \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_\sigma} \left[\frac{1}{2} \left\| s_\phi(\tilde{\boldsymbol{x}}; \sigma) \right\|^2 + \mathrm{div}_{\tilde{\boldsymbol{x}}} \, s_\phi(\tilde{\boldsymbol{x}}; \sigma) \right].$$

Takeaway: SURE $\leftrightarrow$ denoising risk $\leftrightarrow$ score matching $\Rightarrow$ DSM is a denoising principle in disguise.

Summary: From SM to Denoising SM

- **Why Score Matching?** ✓
- **Why Denoising Score Matching (DSM)?** ✓
- **What techniques can further improve DSM?** ✓
- **What can we do when clean data are unavailable?** ✓

Discussion Questions?

1. What do we gain from higher-order scores? When is it worth the complexity?
2. Others?

References

Lai, Chieh-Hsin, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. "The principles of diffusion models." arXiv preprint arXiv:2510.21890 (2025).

Pascal Vincent. *A Connection Between Score Matching and Denoising Autoencoders*. Tech Report, 2011.

Chenlin Meng et al. *Estimating High Order Gradients of the Data Distribution by Denoising*. arXiv:2111.04726, 2021.