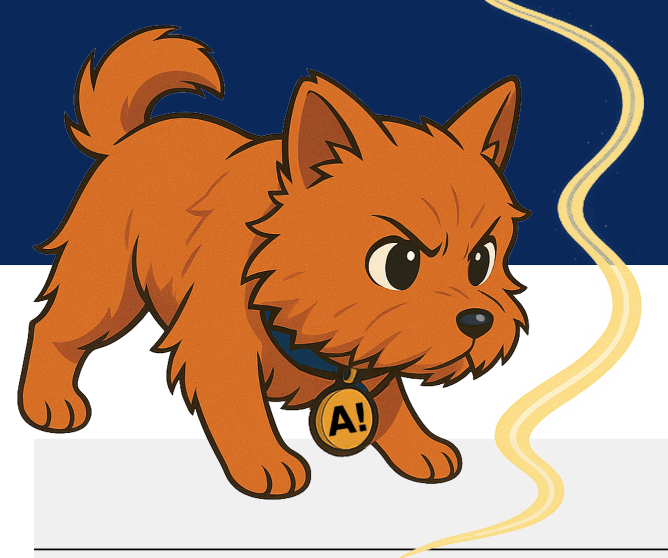
Free Hunch: Denoiser Covariance Estimation for Diffusion Models Without Extra Costs

Severi Rissanen Markus Heinonen Arno Solin

Department of Computer Science, Aalto University

Aalto University



TL;DR

- We propose **Free Hunch (FH)**: a training-free method to estimate denoiser covariances in diffusion models.
- FH combines **data covariances** and **trajectory curvature** to provide accurate guidance.
- FH enables strong results in conditional generation tasks like image deblurring, even with few solver steps.

Background

- The diffusion model score conditional on a condition $\mathbf{y}$ can be composed as:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t), \quad (1)$$

- the conditional score can be calculated with:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \int \underbrace{p(\mathbf{y} \mid \mathbf{x}_0)}_{\text{constraint}} \underbrace{p(\mathbf{x}_0 \mid \mathbf{x}_t)}_{\text{denoise}} \, d\mathbf{x}_0 = \nabla_{\mathbf{x}_t} \log \mathbb{E}_{p(\mathbf{x}_0 \mid \mathbf{x}_t)} \left[p(\mathbf{y} \mid \mathbf{x}_0) \right]. \quad (2)$$

- The posterior $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ is difficult. Common approach: Gaussian $p(\mathbf{x}_0 \mid \mathbf{x}_t) \approx \mathcal{N}(\mathbf{x}_0 \mid \mu_{0\mid t}(\mathbf{x}_t), \mathbf{\Sigma}_{0\mid t}(\mathbf{x}_t))$. The mean comes from the denoiser, but the **covariance is hard**.
- Existing methods require extra training or approximations (heuristics, Jacobians).
- Free Hunch estimates covariance from:
  - **Data covariance** (from training samples)
  - **Curvature along the generative trajectory** (via Tweedie)
- Accurate covariance ⇒ better guidance ⇒ better results.

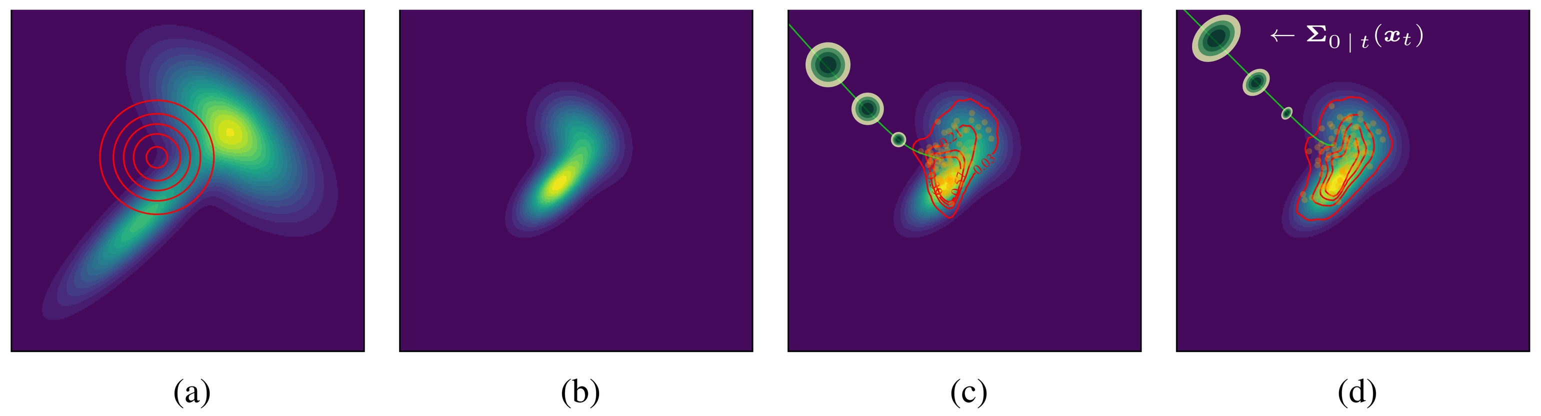


Figure 1. (a) A distribution $p(\mathbf{x}_0)$ represented by a pretrained diffusion model, and a Gaussian likelihood $p(\mathbf{y} \mid \mathbf{x}_0)$. (b) The (exact) posterior $p(\mathbf{x}_0 \mid \mathbf{y}) \sim p(\mathbf{x}_0)p(\mathbf{y} \mid \mathbf{x}_0)$. (c) Generated samples from a model with a heuristic diagonal denoiser covariance $\mathbf{\Sigma}_{0\mid t}(\mathbf{x}_t)$, and a generative ODE trajectory with approximated $p(\mathbf{x}_0 \mid \mathbf{x}_t)$ shapes represented as ellipses along the trajectory. (d) Generated samples with our denoiser covariance.

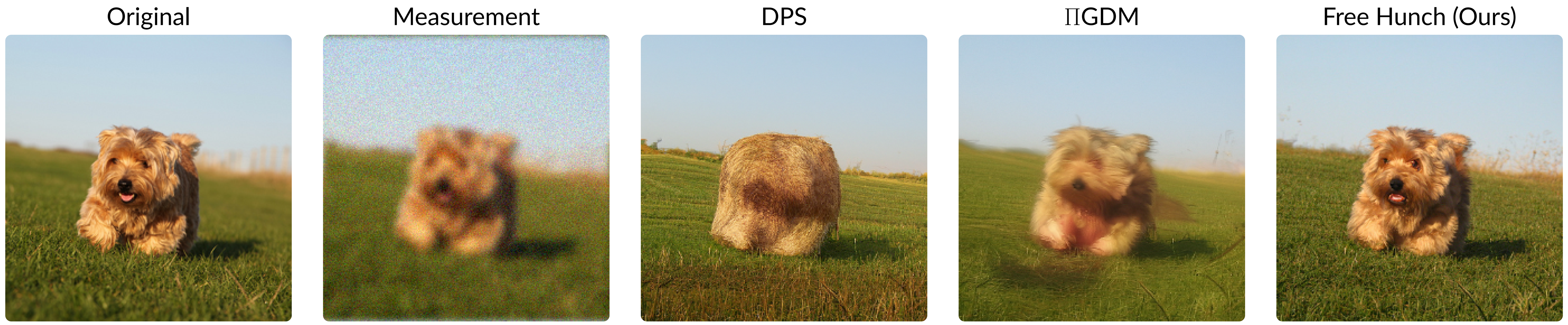


Figure 2. Comparison of different conditional diffusion methods for deblurring, with a low number of solver steps (15 Heun iterations). DPS [?] and ΠGDM [?] work well with many steps, but accurate covariance estimates matter more for small step counts.

Method

- FH estimates $Cov[\mathbf{x}_0 \mid \mathbf{x}_t]$ using:
  - **Time updates:** Approximately transfer estimates across noise levels with a second-order approximation on $\log p(\mathbf{x}_t)$.
  - **Space updates:** BFGS-style low-rank updates during sampling.
- Efficient structure:
  - $\mathbf{\Sigma} = D + UU^\top - VV^\top$ (diagonal + low-rank)
- No retraining. No Jacobians. Works with any sampler.
- Initialized using DCT-diagonal data covariances.

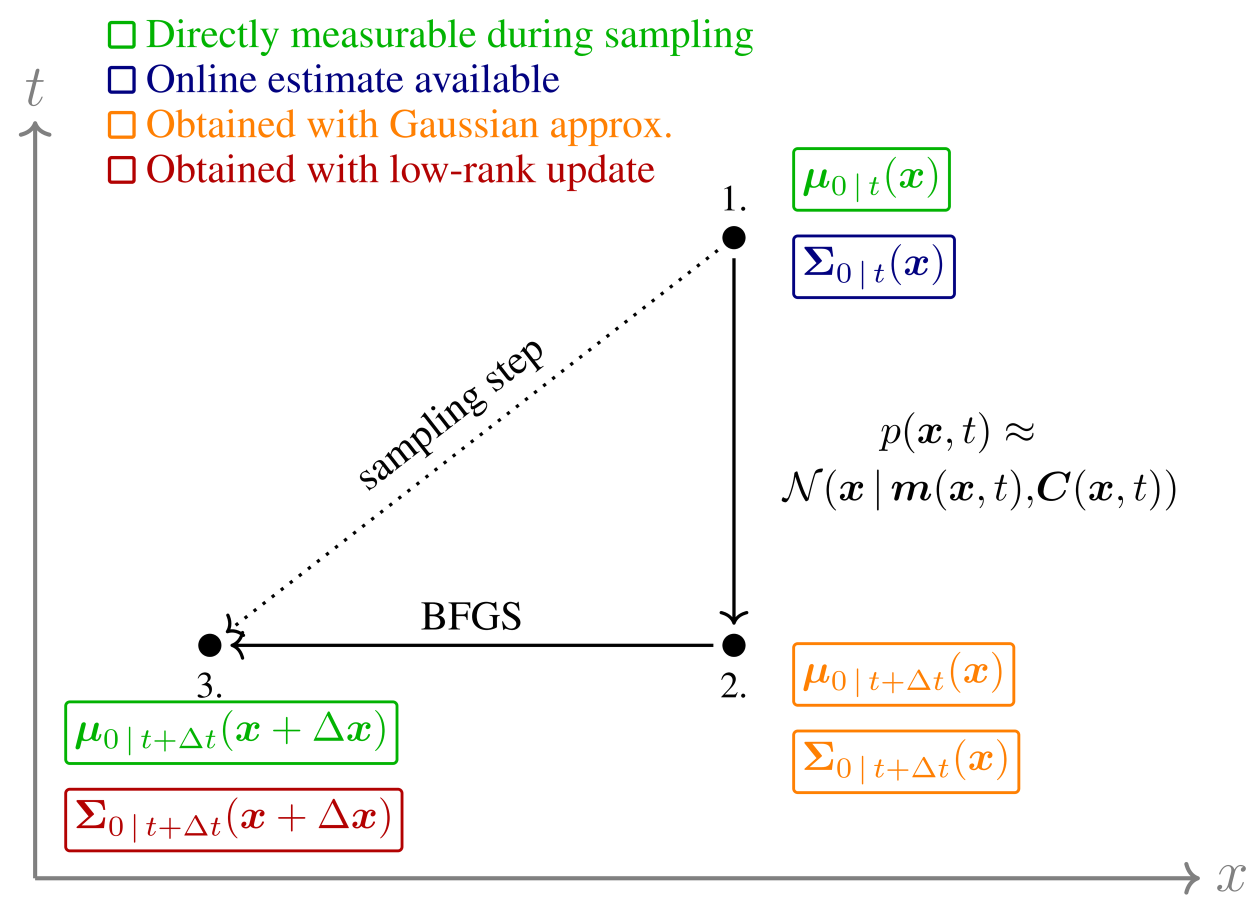


Figure 3. Sketch of our method during sampling.

Results

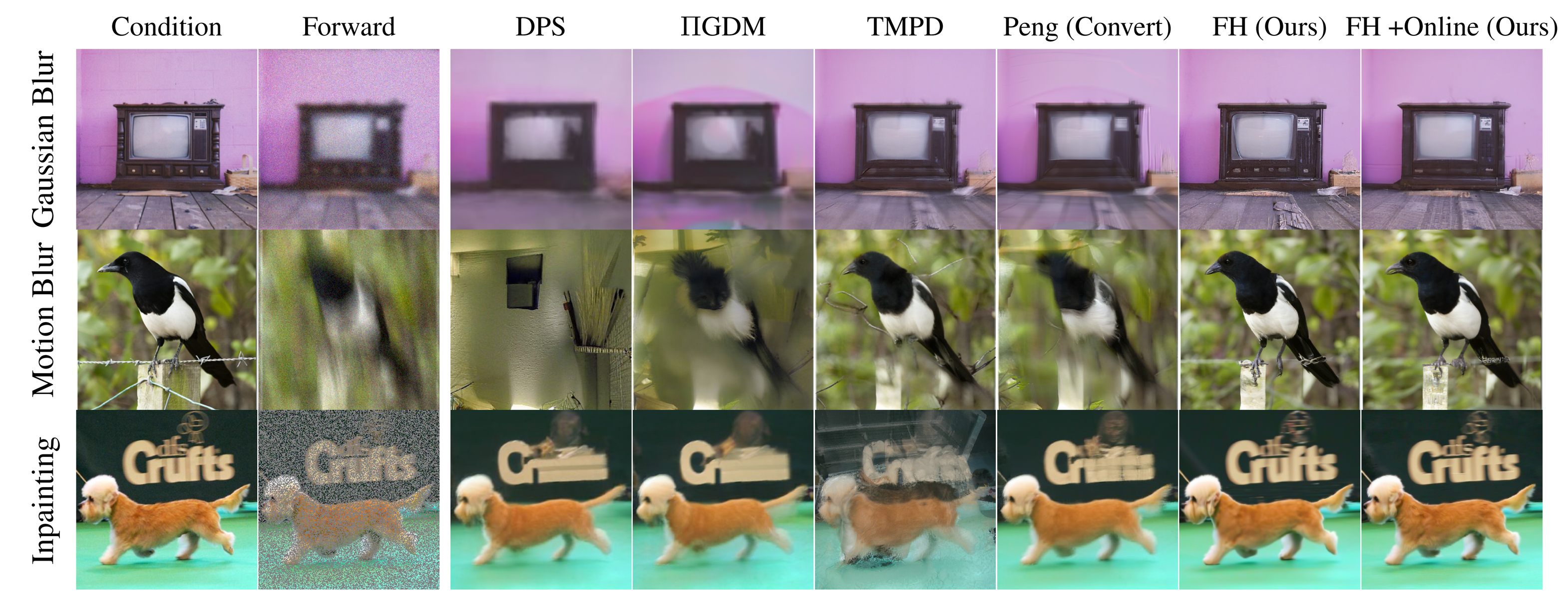


Figure 4. Qualitative examples using the 15-step Heun sampler for image restoration methods for deblurring (Gaussian), inpainting (random) and deblurring (motion).

	Method	Deblur (Gaussian)			Inpainting (Random)			Deblur (Motion)			Super res. (4×)		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
15 steps	DPS	19.94	0.444	0.572	20.68	0.494	0.574	17.02	0.354	0.646	19.85	0.460	0.590
	ΠGDM	20.29	0.474	0.574	19.87	0.468	0.598	19.21	0.429	0.602	20.17	0.474	0.582
	TMPD	22.56	0.572	0.486	17.70	0.447	0.589	20.40	0.481	0.567	21.15	0.517	0.541
	Peng Convert	22.53	0.563	0.490	22.23	0.579	0.489	20.46	0.475	0.556	21.92	0.541	0.517
	Peng Analytic	22.52	0.563	0.490	22.14	0.574	0.494	20.46	0.475	0.556	21.92	0.541	0.517
	DDNM+	7.21	0.029	0.822	23.95	0.667	0.352	–	–	–	**24.30**	**0.669**	0.398
	DiffPIR	22.77	0.575	0.403	16.10	0.284	0.661	19.75	0.381	0.527	21.76	0.540	0.436
	Identity	22.91	0.594	0.384	18.83	0.397	0.590	20.06	0.393	0.506	22.65	0.589	0.412
	Identity+online	23.08	0.606	0.385	18.86	0.397	0.590	20.31	0.418	0.492	22.76	0.597	0.414
	FH	23.41	0.625	**0.373**	24.76	0.702	0.327	21.69	0.534	0.447	23.39	0.632	**0.390**
	FH+online	**23.57**	**0.635**	0.378	**25.29**	**0.731**	**0.315**	**21.83**	**0.548**	**0.442**	23.31	0.624	0.393

Figure 5. Results with the **Euler solver**. Our model performs especially well at small step sizes and remains competitive at larger step counts as well.

severi.rissanen@aalto.fi