Laplacian Score Sharpening for Mitigating Hallucination in Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models, though successful, are known to suffer from hallucinations that create incoherent or unrealistic samples. Recent works have attributed this to the phenomenon of mode interpolation and score smoothening, but they lack a method to prevent their generation during sampling. In this paper, we propose a post-hoc adjustment to the score function during inference that leverages the Laplacian (or sharpness) of the score to reduce mode interpolation hallucination in unconditional diffusion models across 1D, 2D, and high-dimensional image data. We derive an efficient Laplacian approximation for higher dimensions using a finite-difference variant of the Hutchinson trace estimator. We show that this correction significantly reduces the rate of hallucinated samples across toy 1D/2D distributions and a high-dimensional image dataset. Furthermore, our analysis explores the relationship between the Laplacian and uncertainty in the score.

1 Introduction

Generative modeling has advanced rapidly, enabling the creation of highly realistic and complex data across diverse fields such as image synthesis, language modeling, and molecular design. Among these approaches, diffusion models have gained prominence by generating samples through the iterative refinement of noisy data [3]. This refinement is guided by the score function—the gradient of the log-probability—which determines the noise to be removed at each timestep and thereby reveals the underlying data distribution [8]. Despite their success, these models still suffer from artifacts and hallucinations, where generated samples contain unrealistic or incoherent elements not present in the training distribution. A common example is the generation of human hands with extra fingers, illustrating a failure to perfectly capture the true data manifold.

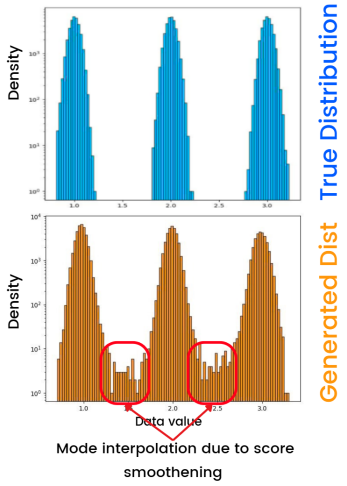


Figure 1: Illustration of the mode interpolation phenomenon in diffusion models, generated samples fall in between the true data modes, resulting in hallucinations

A key study by Aithal et al. [1] identifies mode interpolation as the primary source of such hallucinations in unconditional image generation as shown in Figure 1, attributing it to an excessive smoothing of the score function in inter-mode regions. This work directly motivates our goal of targeting these regions of uncertain score estimates. In a complementary work, Jeon et al. [5] analyze the curvature of the log-probability (the trace of the score's first derivative) to quantify overfitting and mitigate memorization of training samples in diffusion models. This concept of leveraging curvature is supported by Lee and Park [6], who introduced

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

curvature-based regularization using Hutchinson's trace estimator [4] for efficient training of Autoencoders. While the memorization study [5] focuses on the score and its Jacobian (first derivative), we hypothesize that the Laplacian (the trace of the Hessian, or second derivative) provides a crucial signal for uncertainty in the score function. This leads us to ask:

- Can the Laplacian be leveraged to enhance the score in inter-mode regions, thereby mitigating hallucinations?
- Can the Laplacian also serve as an indicator of regions where the model exhibits uncertainty in its estimated score?

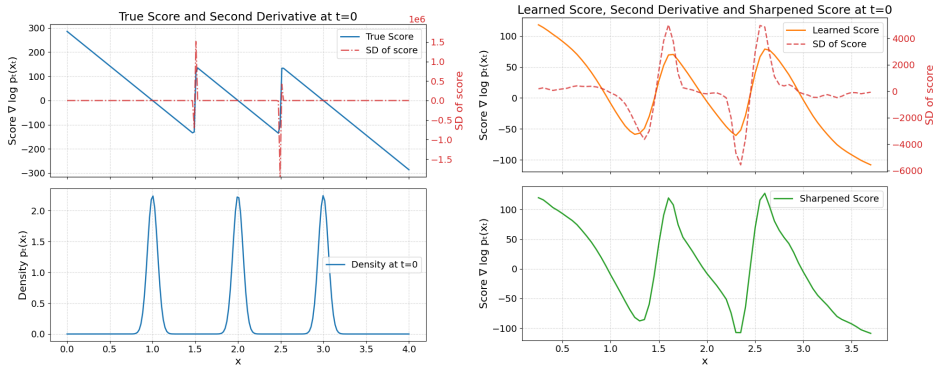
2 Background

The core of the diffusion process is guided by the score function [7], defined as the gradient of the log-probability density, $\nabla_x \log p(x)$. This vector field points in the direction of steepest ascent in the data probability density. During sampling, this effectively acts as a guide, indicating the path from noisy inputs toward cleaner data. In practice, the model learns to estimate this score through the denoising objective of the diffusion process. A key result [8] establishes that the noise ϵ predicted by a DDPM is a scaled estimate of this score for the noisy distribution at timestep t:

$$\epsilon_\theta(x_t, t) \propto -\nabla_{x_t} \log p_t(x_t)$$

The Jacobian(first derivative) of the score function is equivalent to the Hessian of the log-probability, $\nabla^2 \log p(\mathbf{x})$. The properties of this matrix denote the local curvature of the data distribution. Regions with high curvature (e.g., a large negative trace) indicate high local confidence, while regions with low curvature (trace near zero) indicate low confidence [5]. This is clearly observed when a generated sample $\mathbf{x}$ is near a training sample where the log-probability surface is sharply peaked, resulting in a large negative trace of high magnitude. Consequently, the trace has been proposed as a regularizer to avoid memorization of training samples.

Building on this, we hypothesize that the second derivative of the score captures abrupt changes in the confidence—highlighting regions where the model is uncertain about the correct trajectory. As shown in Figure 2, both the true and learned scores exhibit non-zero second-order derivatives primarily in inter-mode regions, while remaining near zero at the modes themselves. However, the learned score is overly smoothed due to the model's inability to represent sharp transitions [1], resulting in a diminished signal near inter-modes and causing sample trajectories to stagnate. To address this, we aim to sharpen the score, selectively increasing its magnitude in these uncertain regions to guide sample trajectories away from them and mitigate mode-interpolation hallucinations.



(a) (b)

Figure 2: (a) A toy 1D distribution with modes at $x = \{1, 2, 3\}$ along with its true score function (blue) and the second derivative of the true score. (b) A visualization of score sharpening in the 1D scenario, from top to bottom: the learned score function (orange) corresponding to the distribution in (a), the second derivative (Laplacian) of the learned score, and the new sharpened score function (green).

3 Methodology

3.1 Peak Sharpening with Second-Order Derivative

Building on this observation, peak sharpening directly leverages the second derivative to enhance the peaks of the score function. Specifically, we perform a weighted subtraction of the original function and its second derivative:

$$R_j = Y_j - \alpha Y_j'' \tag{1}$$

where R_j is the sharpened (peak-enhanced) score, Y_j is the original score, and Y_j'' denotes its second derivative.

To reduce computational overhead, we approximate the Second order derivative using the finite difference method at various points. This approximation can be seamlessly integrated into the sampling process of a DDPM (or, more broadly, any score-based diffusion model) to improve sample quality and mitigate hallucinations. Figure **??** illustrates the effect of the sharpening operation on a 1D score function. After peak enhancement, the score function (green) exhibits sharper transitions at inter-modes compared to the learned score (orange), while remaining unchanged elsewhere. Algorithm 1 outlines the sampling procedure for the 1D scenario with finite difference approximation for the second derivative. For 1D (and 2D setting) the score sharpening algorithm involves three hyperparameters:

- **Timestep threshold** ($t_{\text{threshold}}$): The forward timestep defining the start of sharpening in the reverse process. A higher $t_{\text{threshold}}$ means sharpening is applied for *more* reverse steps, including *noisier* states. This can improve peak recovery but increases artifact risk. Sharpening is ineffective in very noisy states (high forward t), as the score lacks sharp peaks at the inter-modes.(Appendix A.1)
- **Perturbation size** (δ): The finite-difference step size used to estimate the second derivative. Smaller δ values produce sharper but noisier estimates, while larger δ values yield smoother but less precise curvature information.
- **Enhancement strength** (α): The scaling factor applied to the negative second derivative during sharpening. Lower α values result in minimal sharpening, whereas higher α values can exaggerate peaks and introduce artifacts.

Algorithm 1 Sampling with Score Sharpening in 1D setting

function SHARPENED_DENOISE($x, t,$ denoise_fn$, \delta, \alpha, t_{\text{threshold}}$)
  $f_x \leftarrow$ denoise_fn(x, t) ▷ f_x estimates noise ϵ at timestep t
  if $t < t_{\text{threshold}}$ **then** ▷ $t_{\text{threshold}}$: timestep threshold
    $f_x^+ \leftarrow$ denoise_fn$(x + \delta, t)$ ▷ δ: perturbation size
    $f_x^- \leftarrow$ denoise_fn$(x - \delta, t)$
    second_derivative $\leftarrow \frac{f_x^+ + f_x^- - 2f_x}{\delta^2}$
    $f_x \leftarrow f_x - \alpha \cdot$ second_derivative ▷ α: sharpening strength
  end if
  return f_x
end function

for $t = T, T-1, \ldots, 1$:
  $x \leftarrow x -$ Sharpened_Denoise$(x, t,$ denoise_fn$, \delta, \alpha, t_{\text{threshold}})$

3.2 Estimating Laplacian in Higher Dimensions

In a 1D distribution, the second derivative of the score can be computed by applying small perturbations in the forward and backward directions. In higher dimensions, this generalizes to the Laplacian, which represents the trace of the Hessian of the score. For instance, in 2D, the Laplacian can be approximated by applying perturbations along both dimensions, requiring four forward calls.

$$\text{Laplacian}_{2D}(f) \approx \frac{f(x+\delta, y) + f(x-\delta, y) + f(x, y+\delta) + f(x, y-\delta) - 4f(x, y)}{\delta^2}$$

However, this direct extension becomes computationally expensive for high-dimensional inputs such as images. To overcome this, we employ a finite-difference variant of the Hutchinson trace estimator, enabling efficient Laplacian estimation for each pixel. Crucially, the score function $s(\mathbf{x}) \in \mathbb{R}^d$ is a vector unlike in the 1D setting; its output consists of d scalar components $s_k(\mathbf{x})$, each representing the score for a single pixel. The Hessian of the entire score function is therefore a third-order tensor. However, for our sharpening method, we require only the Laplacian for each output component s_k—that is, the trace of the Hessian $\mathbf{H}_k$ of each scalar function $s_k(\mathbf{x})$. The Hutchinson identity states that for any such Hessian matrix $H_k \in \mathbb{R}^{d\times d}$

$$\mathrm{tr}(H_k) = \mathbb{E}_v\left[v^\top H_k v\right], \tag{2}$$

where v is a random Rademacher vector (i.e., each entry independently takes values ± 1 with equal probability). $v^\top H_k v$ in Eq. 2 denotes the second directional derivative of s_k along the direction v. Using a central finite-difference approximation, it can be estimated as

$$v^\top H_k v \approx \frac{s_k(x+\delta v) - 2s_k(x) + s_k(x-\delta v)}{\delta^2}, \tag{3}$$

where δ is a small perturbation. Taking the expectation of Eq. 3 over Rademacher vectors v, the Laplacian of s_k can be given as

$$Laplacian_k = \mathrm{tr}(H_k) \approx \mathbb{E}_v\left[\frac{s_k(x+\delta v) - 2s_k(x) + s_k(x-\delta v)}{\delta^2}\right]. \tag{4}$$

This computation is performed for all pixels simultaneously by evaluating the score $s(x \pm \delta v)$. This yields a vector $L(\mathbf{x}) \in \mathbb{R}^d$ where the k-th entry is the Laplacian of s_k. This vector $L(\mathbf{x})$ will serve the function of Y_j'' in Eq. 1. We replace the second derivative in Algorithm 1 with the $L(\mathbf{x})$ for images. This method allows us to efficiently estimate the Laplacian with significantly fewer function evaluations compared to computing second derivatives along each input dimension explicitly. The algorithm for sampling in images, including hyperparameter choices, is detailed in A.2

4 Experiments

4.1 1D and 2D Synthetic Data

We quantitatively evaluate our sharpening method on synthetic 1D and 2D distributions to measure its ability to reduce hallucinations while preserving distributional fidelity. We train a DDPM using a linear noise schedule over 1000 timesteps for 1000 epochs on the denoising objective. Hallucinated samples are defined as those lying beyond a predefined distance threshold from the modes of the training distribution. We report the Intermodulation Count (IM Count), which counts such inter-mode samples, and the L1 norm between the generated and target distributions at the end of 1000 epochs to assess overall fidelity.

For our sharpening method, we use a timestep threshold of $t_{\text{threshold}} = 50$, a perturbation scale of $\delta = 0.1$ (1D) and $\delta = 0.05$ (2D), and a sharpening strength of $\alpha = 0.01$ (1D) and $\alpha = 0.0025$ (2D). The values for α were chosen based on the empirical ratio between the score and its second derivative.

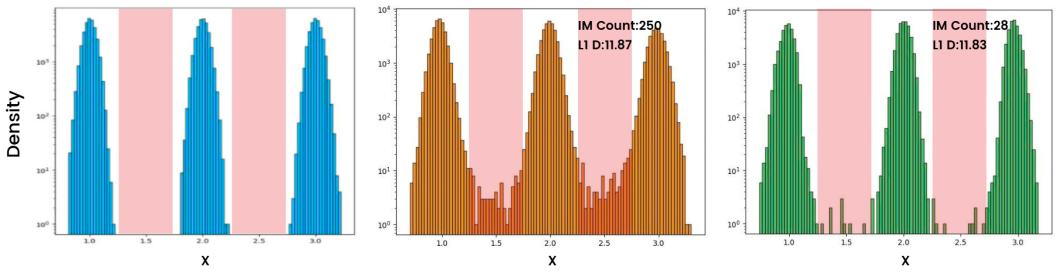


Figure 3: Histograms of the true 1D distribution (blue) and generated distributions using the vanilla (orange) and sharpened (green) scores for 100,000 samples. Sharpening the score reduces the number of samples generated in inter-mode regions.

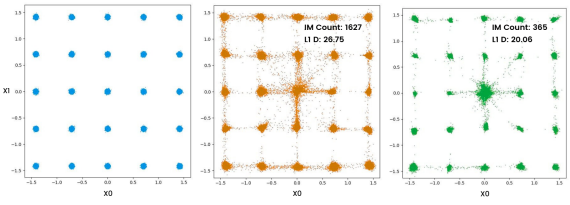


Figure 4: A Scatter Plot of the true 2D data distribution for 100,000 samples is shown in blue. Samples generated by the vanilla model (orange) exhibit grid-like interpolation artifacts between modes, which are reduced in samples generated using our sharpened score (red).

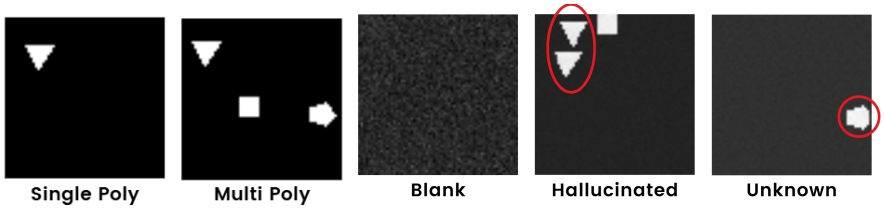
Table 1: Image Quality Distribution: Vanilla vs. Sharpened Score for 1000 images. We see hallucinated images reduce by 3.9 percentage points and blank images increase by 4 percentage points.

Image Type	Vanilla (%)	Sharpened (%)
Hallucinated	6.00	2.10
Unknown Shapes	0.60	0.60
Blank Images	1.30	5.30
Good Images	92.10	92.00

After applying the sharpened score during inference, we observe a significant decrease in the IM Count, indicating a reduction in hallucinated samples. For instance, in the 1D case (Figure 3), the IM Count decreased from 250 to 28, while in 2D (Figure 4), it reduced from 1627 to 365. Crucially, the L1 norm either remains unchanged or shows a slight reduction, demonstrating that the core structure of the distribution is preserved while artifacts are mitigated.

4.2 Rule-Based Toy Dataset: Shapes

For image-based evaluation, we use the rule-based Shapes dataset introduced by [1] to quantitatively measure hallucinations. The training set always contains a single instance of each type of polygon—triangle, square, and pentagon—without repetition of the same shape. An image is classified as: **Good Image:** Contains one or more polygons with no repetition among the three valid shapes. **Hallucinated Image:** Contains more than one polygon of the same shape (repetition). **Unknown Shape:** Contains a polygon that is not a triangle, square, or pentagon. **Blank Image:** Contains no shape and is predominantly noisy.



(a) (b)

Figure 5: (a) Samples on the true data manifold of Shapes Dataset, (b) Samples generated by the diffusion model, which are not part of the true data manifold.

We train a unconditional DDPM with UNET architecture using a cosine noise scheduler similar to ADM [2] over 1000 timesteps for 50 epochs. After applying our sharpening method during inference, we observe a strong reduction in mode interpolation hallucinations as shown in Table 1. The proportion of hallucinated images dropped from 6.0% to 2.1%, affirming that sharpening effectively guides samples away from the inter-mode regions. However, this intervention also resulted in a corresponding increase in blank images from 1.3% to 5.3%, suggesting that while sharpening pushes samples away from inter-mode regions, it does not subsequently guide them toward the true data manifold as effectively as it does in 1D or 2D settings. The proportions of good and unknown-shape images remained stable, indicating that the core generative capability is preserved.

On plotting the mean of the top 100 absolute Laplacian magnitudes in $|L(\mathbf{x})|$ corresponding to each pixel (Figure 6, Middle), we observed a clear peak in the mean between 600 and 800 timesteps (in ascending order of sampling) for the majority of hallucinated samples. Interestingly, this peak was not exclusive to images with final hallucinations; it also appeared in a subset of ultimately clean samples that exhibited transient artifacts— hallucinated structures that emerged mid-sampling but were later corrected. Furthermore, when we mapped the pixels corresponding to these top 100 values

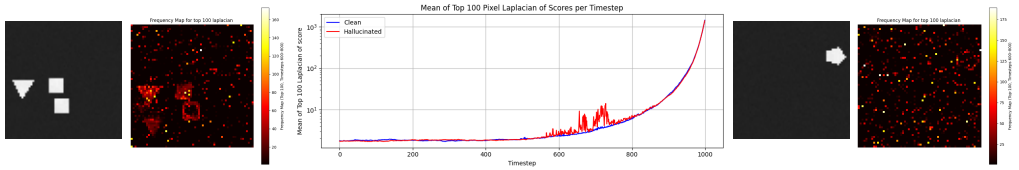


Figure 6: **Left**: Frequency map of top 100 absolute Laplacian magnitudes (timesteps 600–800) for a hallucinated image (red curve, middle). **Middle**: Mean of the top 100 absolute Laplacian magnitudes over time. **Right**: Frequency map for a clean image (blue curve, middle).

on a frequency map(Figure 6, Left & Right), we found that these high values were associated with hallucinated shapes—either those appearing in the final image or those that emerged during the image trajectory but were absent in the final image.

5 Conclusion and Future Work

In this work, we introduced a novel, post-hoc sharpening technique for the score function using its Laplacian to reduce hallucinations. We derived an efficient method to approximate the high-dimensional Laplacian using a finite-difference variant of the Hutchinson estimator, making our method scalable to images. We provided an analysis linking the Laplacian of the score to uncertainty and hallucination during the sampling process, validated for 1D setting and the image dataset-Shapes.

Though the method is able to reduce hallucinated samples in the high dimensional Shapes dataset, the core limitation of this corrective approach is its destructive nature. Consequently, it is ideal for pruning errors but less suitable for open-ended discovery tasks like de novo protein generation, where the goal is to explore and refine uncertain regions into viable solutions. An additional limitation is its sensitivity to hyperparameters, such as the sharpening strength α and the perturbation scale δ, which can be challenging to tune optimally. Future work will focus on integrating the Laplacian signal more adaptively into the sampling process to not only avoid artifacts but also proactively guide samples toward the data manifold. We will also explore methods to automate the selection of α and δ to improve the algorithm's practicality.

References

[1] Sumukh K Aithal, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation, 2024.

[2] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020.

[4] Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990.

[5] Dongjae Jeon, Dueun Kim, and Albert No. Understanding and mitigating memorization in generative models via sharpness of probability landscapes, 2025.

[6] Yonghyeon Lee and Frank Chongwoo Park. On explicit curvature regularization in deep generative models, 2023.

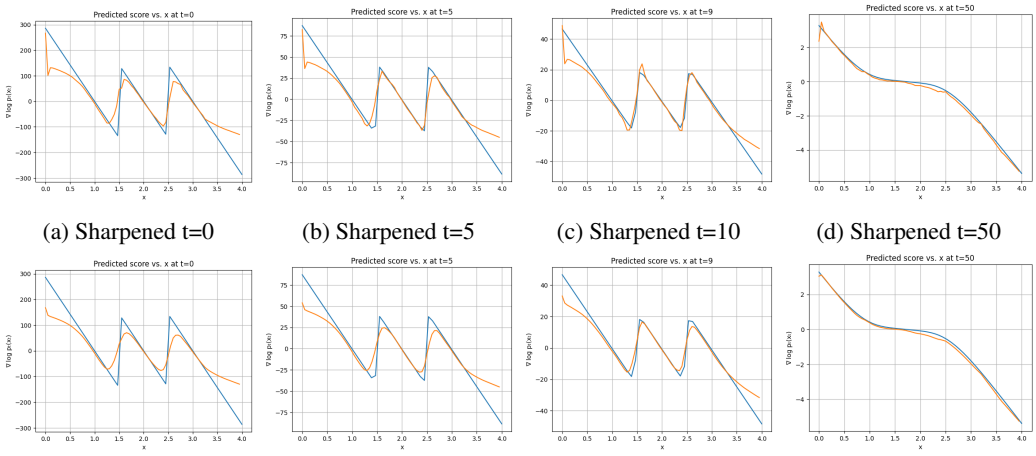
[7] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[8] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

A Technical Appendices and Supplementary Material

A.1 Evolution of Score over the Timesteps

Figure 7 illustrates the evolution of the score function in the 1D setting, comparing the learned score with and without sharpening. We observe that the sharpened score aligns more closely with the true score than the learned (vanilla) score up to approximately timestep 50 (corresponding to a late stage in the reverse diffusion where noise is low), effectively recovering sharper mode boundaries. Beyond timestep 50, however, the true score no longer exhibits pronounced peaks, and applying sharpening in this regime may introduce unnecessary artifacts rather than improving alignment.



(a) Sharpened t=0 (b) Sharpened t=5 (c) Sharpened t=10 (d) Sharpened t=50

(e) Vanilla t=0 (f) Vanilla t=5 (g) Vanilla t=10 (h) Vanilla t=50

Figure 7: Comparison of sharp vs vanilla scores across different timesteps (forward).

A.2 Laplacian sharpening Algorithm for Images

For the Hutchinson estimation, $n_{samples}$ specifies the number of Rademacher vectors sampled to approximate the second derivative. In our experiments, we set $n_{samples}$= 3. Unlike in 1D or 2D, we found that applying score sharpening during early timesteps is ineffective. This is because the model primarily learns finer details in the early timesteps, whereas hallucinations tend to appear at later timesteps. To address this, we introduce a timestep range for sharpening, with a lower bound to avoid early timesteps. Specifically, we used the range (200, 400) in the forward diffusion convention, or (600, 800) in the sampling convention, as inferred from Figure 6, Middle. We use $\delta = 0.05$ and $\alpha = 0.05$ with α being inferred based on the observed ratio between the score and its second derivative between 600 and 800 timesteps.

Algorithm 2 Score Sharpening with Rademacher Perturbations

function SHARPENED_DENOISE($x, t, t_{\text{low}}, t_{\text{high}}$, denoise_fn, $\delta, \alpha, n_{\text{samples}}$)
  $f_x \leftarrow$ denoise_fn(x, t) ▷ estimates noise vector ϵ at timestep t
  if $t_{\text{low}} < t < t_{\text{high}}$ **then** ▷ $(t_{\text{low}}, t_{\text{high}})$: timestep range
    Laplacian_total $\leftarrow 0$
    for $i = 1$ **to** n_{samples} **do**
      $r \leftarrow$ Rademacher(x) ▷ Random ±1 perturbation per element
      $h_i \leftarrow \delta \cdot r$ ▷ δ: perturbation size
      $f_x^+ \leftarrow$ denoise_fn$(x + h_i, t)$
      $f_x^- \leftarrow$ denoise_fn$(x - h_i, t)$
      Laplacian_total $\leftarrow$ Laplacian_total $+ \frac{f_x^+ - 2f_x + f_x^-}{h_i^2}$
    end for
    Laplacian $\leftarrow$ Laplacian_total$/n_{\text{samples}}$
    $f_x \leftarrow f_x - \alpha \cdot$ Laplacian ▷ α: sharpening strength
  end if
  return f_x
end function