

1 **Part I**

2 **Appendix**

3 **Table of Contents**

5	A Derivations and proofs	11
6	A.1 Optimal denoiser: derivation, equivalence of ϵ and x_0 parametrization	11
7	A.2 Patch-based optimal denoiser: formal derivation	12
8	A.3 Ours: why do we binarize the sensitivity field	14
9	A.4 “Pass-through” denoisers: detailed analysis of SNR	16
10	A.5 Manipulating the sensitivity field: variance of the perturbation	17
11	B Additional Experiments and Ablation	17
12	B.1 Ablation of our model	17
13	B.2 Self-attention layers in denoising U-Nets	18
14	B.3 How to reproduce the reported sensitivity fields	18
15	B.4 Sensitivity field of the optimal denoiser	19
16	B.5 Generation dynamics	20
17	B.6 Variance of the reported metrics	21
18	B.7 Quantitative measure of novelty of samples	21
19	B.8 Additional generation results	22
20	C Implementation details	22
21	C.1 Sampling.	22
22	C.2 Training DDPM Model	22
23	C.3 Our analytical model	22
24	C.4 Baseline implementation details	23
25	C.5 Computational resources and runtime	24

29 A Derivations and proofs

30 In this section, we provide detailed derivations and proofs for the background and the claims made in
31 the paper.

32 A.1 Optimal denoiser: derivation, equivalence of ϵ and x_0 parametrization

33 We begin by defining the optimal denoiser for the x_0 parameterization we use in the paper.

34 **Definition A.1.** *The optimal denoiser $\hat{f}(x, t)$ for a data distribution X at a particular noise level t is*
35 *the minimizer of the loss function*

$$\min_f \mathbb{E}_{\substack{x_0 \sim X \\ \epsilon \sim N(0, I)}} \|f(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) - x_0\|_2^2 \quad (1)$$

36 Recall that $\sigma_t^2 = 1 - \alpha_t$.

37 **Proposition A.2.** *When $X = \{x_0^i\}_{i \in [N]}$ is a finite empirical distribution, the optimal denoiser*
38 *$\hat{f}(x, t)$ has the following analytical expression:*

$$\hat{f}(x, t) = \sum_i x_0^i \operatorname{softmax}_i \left\{ -\|x - \sqrt{\alpha_t}x_0^j\|^2 / 2\sigma_t^2 \right\}. \quad (2)$$

39 *Proof.* We first write down the objective (1) in terms of the random variable $x = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon$:

$$\begin{aligned} & \mathbb{E}_{\substack{x_0 \sim X \\ \epsilon \sim N(0, I)}} \left[\|f(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) - x_0\|_2^2 \right] \\ &= \mathbb{E}_{\substack{x_0 \sim X \\ x \sim N(\sqrt{\alpha_t}x_0, \sigma_t I)}} \|f(x, t) - x_0\|_2^2 \\ &= \int \mathbb{E}_{x_0 \sim X} \left[\exp\left(-\|\sqrt{\alpha_t}x_0 - x\|^2 / 2\sigma_t^2\right) \|f(x, t) - x_0\|_2^2 \right] dx \end{aligned}$$

40 We then minimize the integral coordinate-wise for each x to get the optimal $f(x, t)$:

$$\begin{aligned} 0 &= \mathbb{E}_{x_0 \sim X} \left[\exp\left(-\|\sqrt{\alpha_t}x_0 - x\|^2 / 2\sigma_t^2\right) (\hat{f}(x, t) - x_0) \right] \\ \hat{f}(x, t) &= \frac{\sum_i x_0^i \exp(-\|x - \sqrt{\alpha_t}x_0^i\|^2 / 2\sigma_t^2)}{\sum_j \exp(-\|x - \sqrt{\alpha_t}x_0^j\|^2 / 2\sigma_t^2)}. \end{aligned}$$

41 Using the definition of $\operatorname{softmax}_i \{a_j\}_{j \in [N]} = \frac{\exp(a_i)}{\sum_{j=1}^N \exp(a_j)}$, we get (2).

42 □

43 **Definition A.3.** *The optimal denoiser $\hat{\epsilon}(z, t)$ for a data distribution X at a particular noise level t is*
44 *the minimizer of the loss function*

$$\min_f \mathbb{E}_{\substack{x_0 \sim X \\ \epsilon \sim N(0, I)}} \|f(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) - \epsilon\|_2^2 \quad (3)$$

45 **Proposition A.4.** *When $X = \{x_0^i\}_{i \in [N]}$ is a finite empirical distribution, the optimal denoiser for*
46 *the ϵ -parameterization $\hat{\epsilon}(x, t)$ can be written in terms of that of the x -parameterization $\hat{f}(x, t)$:*

$$\hat{\epsilon}(x, t) = (x - \sqrt{\alpha_t}\hat{f}(x, t)) / \sigma_t \quad (4)$$

47 *Proof.* We follow the same proof as Proposition A.2, with the main difference being the following
 48 step:

$$\begin{aligned} & \mathbb{E}_{\substack{x_0 \sim X \\ \epsilon \sim N(0, I)}} \left[\|f(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) - \epsilon\|_2^2 \right] \\ &= \mathbb{E}_{\substack{x_0 \sim X \\ x \sim N(\sqrt{\alpha_t}x_0, \sigma_t I)}} \|f(x, t) - (x - \sqrt{\alpha_t}x_0)/\sigma_t\|_2^2. \end{aligned}$$

49

□

50 **Remark A.5.** Another way to prove Proposition A.2 and Proposition A.4 is to show that the optimal
 51 solutions are of the form $\mathbb{E}[x_0 | x]$ and $\mathbb{E}[\epsilon | x]$, where $x \sim N(\sqrt{\alpha_t}x_0, \sigma_t I)$. Then it becomes clear
 52 that the two expressions are linearly related to each other.

53 A.2 Patch-based optimal denoiser: formal derivation

54 We now turn to the patch-based denoiser, incorporating both locality and equivariance constraints
 55 into the optimal denoising problem as suggested in [2], repeating the derivations in the notations of
 56 this manuscript. Let $X = \{x_0^i\}_{i=1}^N$ be a finite empirical distribution of images, and let

$$M_t^q : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{p \times p}$$

57 denote the operator that extracts a $p \times p$ patch centered at pixel q .

58 As suggested in [2], we impose two constraints on each patch-wise function f^q :

59 1. *Locality*:

$$f^q(x, t) = f^q(M_t^q x, t).$$

60 2. *Equivariance*: For every 2D translation $g \in T(2)$,

$$f(g \circ x, t) = g \circ f(x, t), \implies f^q(g \circ x, t) = g \circ f^{g^{-1}q}(x, t),$$

61 i.e. denoising commutes with the action of $T(2)$ and relocates patches accordingly.

62 **Definition A.6.** The patch-based optimal denoiser $\hat{f}(x, t)$ for a data distribution X at a particular
 63 noise level t is the minimizer of the loss function

$$\begin{aligned} & \min_f \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I), t \sim [0, 1]} \|f(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t) - x_0\|_2^2 \\ & \text{s.t. } f^q(x, t) = f^q(M_t^q x, t), \quad q = 1, \dots, Q, \quad (\text{locality}) \\ & f(g \circ x, t) = g \circ f(x, t), \quad \forall g \in T(2). \quad (\text{equivariance}) \end{aligned} \quad (5)$$

64 **Proposition A.7** (Patch-based optimal denoiser). Under the empirical distribution $X = \{x_0^i\}_{i=1}^N$,
 65 the minimizer $\{\hat{f}^q\}$ of eq. (5) is given, for each patch location q , by

$$\hat{f}^q(x, t) = \sum_i \sum_{g \in T(2)} (g \circ x_0^i)^q \operatorname{softmax}_i \left(-\frac{1}{2\sigma_t^2} \|M_t^q x - \sqrt{\alpha_t} M_t^q [g \circ x_0^i]\|^2 \right), \quad (6)$$

66 and the full-image denoiser is obtained by reconstructing the final image from the pixels above.

67 *Proof of Patch-based optimal denoiser.* We prove this result in three steps: (1) decomposition into
 68 per-pixel optimization, (2) equivalence of equivariance constraint and data augmentation, and (3)
 69 derivation of the local form.

70 **Step 1: Decomposition into per-pixel optimization.** Let $P_q : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$ denote the operator that
 71 extracts pixel q from an image, and define $H = \sum_{q=1}^Q P_q P_q^T$ where P_q^T places a scalar value at pixel
 72 q and zeros elsewhere. Since pixels are disjoint, $P_i P_j^T = 0$ for $i \neq j$, making $\{P_q P_q^T\}$ orthogonal
 73 projections with $\sum_{q=1}^Q P_q P_q^T = I$.

74 By orthogonality of pixel projections:

$$\begin{aligned}\|f(x) - x_0\|_2^2 &= \left\| \sum_{q=1}^Q P_q P_q^T (f(x) - x_0) \right\|_2^2 \\ &= \sum_{q=1}^Q \|P_q f(x) - P_q x_0\|_2^2 \\ &= \sum_{q=1}^Q |f^q(x) - x_0^q|^2\end{aligned}$$

75 Therefore, the original minimization problem decomposes as:

$$\begin{aligned}&\min_f \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \|f(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon) - x_0\|_2^2 \\ &= \sum_{q=1}^Q \min_{f^q} \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} [f^q(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon) - x_0^q]^2\end{aligned}$$

76 Each pixel can be optimized independently.

77 **Step 2: Equivariance constraint equals data augmentation.** For the q -th pixel problem with
78 equivariance constraint:

$$\begin{aligned}&\min_{f^q} \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} [f^q(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon) - x_0^q]^2 \\ &\text{s.t. } f^q(g \circ x) = (g \circ f(x))^q \quad \forall g \in T(2)\end{aligned}$$

79 The equivariance constraint implies that for any translation g : $f^q(g \circ x) = f^{g^{-1}q}(x)$

80 Now consider the data-augmented problem:

$$\min_{f^q} \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \mathbb{E}_{g \sim T(2)} [f^q(g \circ (\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon)) - (g \circ x_0)^q]^2$$

81 Since translation commutes with the noise addition and $(g \circ x_0)^q = x_0^{g^{-1}q}$, this becomes:

$$\min_{f^q} \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \mathbb{E}_{g \sim T(2)} [f^q(\sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t}(g \circ \epsilon)) - x_0^{g^{-1}q}]^2$$

82 If f^q satisfies the equivariance constraint, then $f^q(g \circ x) = f^{g^{-1}q}(x)$, so:

$$\begin{aligned}\mathbb{E}_{g \sim T(2)} |f^q(g \circ x) - x_0^{g^{-1}q}|^2 &= \mathbb{E}_{g \sim T(2)} [f^{g^{-1}q}(x) - x_0^{g^{-1}q}]^2 \\ &= \mathbb{E}_{r \sim T(2)} [f^r(x) - x_0^r]^2\end{aligned}$$

83 By uniform distribution over translations, this equals $|f^q(x) - x_0^q|^2$. Thus, the constrained problem
84 is equivalent to the unconstrained data-augmented problem.

85 **Step 3: Local form derivation.** From the data augmentation equivalence, the optimal denoiser for
86 pixel q minimizes:

$$\mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \mathbb{E}_{g \sim T(2)} [f^q(\sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t} \epsilon) - (g \circ x_0)^q]^2$$

87 With the locality constraint $f^q(x) = f^q(M_t^q x)$, we have:

$$f^q(\sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t} \epsilon) = f^q(M_t^q [\sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t} \epsilon])$$

88 Let $x = \sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t}\epsilon$. The optimal \hat{f}^q satisfies:

$$0 = \mathbb{E}_{\substack{x_0 \sim X \\ g \sim T(2) \\ \epsilon \sim N(0, I)}} \exp \left(-\frac{\|M_t^q x - \sqrt{\alpha_t} M_t^q (g \circ x_0)\|^2}{2\sigma_t^2} \right) (f^q(M_t^q x) - (g \circ x_0)^q)$$

89 Rearranging:

$$\hat{f}^q(M_t^q x) = \frac{\sum_{i=1}^N \sum_{g \in T(2)} (g \circ x_0^i)^q \exp(-\|M_t^q x - \sqrt{\alpha_t} M_t^q (g \circ x_0^i)\|^2 / 2\sigma_t^2)}{\sum_{j=1}^N \sum_{h \in T(2)} \exp(-\|M_t^q x - \sqrt{\alpha_t} M_t^q (h \circ x_0^j)\|^2 / 2\sigma_t^2)}$$

90 Using the softmax notation:

$$\hat{f}^q(x, t) = \sum_{i=1}^N \sum_{g \in T(2)} (g \circ x_0^i)^q \cdot \text{softmax}_{i,g} \left\{ -\frac{1}{2\sigma_t^2} \|M_t^q x - \sqrt{\alpha_t} M_t^q (g \circ x_0^i)\|^2 \right\}$$

91 This completes the proof. □

92 A.3 Ours: why do we binarize the sensitivity field

93 In this section, we provide justification for our algorithm provided in the main paper. In particular,
94 formally justify why it makes sense to binarize the sensitivity fields into a mask of zeros and ones.

95 At first, we generalize the patch-based optimal denoiser by relaxing the locality constraint. Instead of
96 restricting to patch extraction operators M_t^q , we consider arbitrary linear operators $A_t^q : \mathbb{R}^{d \times d} \rightarrow$
97 $\mathbb{R}^{d \times d}$ that can capture more complex spatial relationships.

98 **Definition A.8** (Generalized masked optimal denoiser). *The generalized masked optimal denoiser*
99 *$\hat{f}(x, t)$ for a data distribution X at noise level t is the minimizer of:*

$$\begin{aligned} \min_f \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \|f(\sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, t) - x_0\|_2^2 & \quad (7) \\ \text{s.t. } f^q(x, t) = f^q(A_t^q x, t), \quad q = 1, \dots, Q, & \quad (\text{generalized locality}) \\ f(g \circ x, t) = g \circ f(x, t), \quad \forall g \in T(2). & \quad (\text{equivariance}) \end{aligned}$$

100 , where A_t^q is an arbitrary linear operator.

101 **Proposition A.9** (Generalized masked optimal denoiser). *Following the decomposition and data*
102 *augmentation equivalence from the previous section, the optimal denoiser for pixel q under the*
103 *generalized locality constraint is:*

$$\hat{f}^q(x, t) = \sum_{i=1}^N \sum_{g \in T(2)} (g \circ x_0^i)^q \cdot \text{softmax}_{i,g} \left\{ -\frac{1}{2} \|A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i)\|_{\Sigma_q^{-1}}^2 \right\} \quad (8)$$

104 where $\Sigma_q = \sigma_t^2 A_t^q (A_t^q)^T$ is the covariance matrix of the noise in the transformed space, and $\|\cdot\|_{\Sigma_q^{-1}}^2$
105 denotes the Mahalanobis distance.

106 *Proof.* Following Steps 1 and 2 from the previous section, we arrive at the per-pixel data-augmented
107 problem:

$$\min_{f^q} \mathbb{E}_{x_0 \sim X, \epsilon \sim N(0, I)} \mathbb{E}_{g \sim T(2)} [f^q(\sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t}\epsilon) - (g \circ x_0)^q]^2$$

108 With the generalized locality constraint $f^q(x) = f^q(A_t^q x)$, let $x = \sqrt{\alpha_t}(g \circ x_0) + \sqrt{1 - \alpha_t}\epsilon$. The
109 transformed variable $A_t^q x$ follows:

$$A_t^q x = \sqrt{\alpha_t} A_t^q (g \circ x_0) + \sqrt{1 - \alpha_t} A_t^q \epsilon$$

110 Since $\epsilon \sim N(0, I)$, we have $A_t^q \epsilon \sim N(0, A_t^q (A_t^q)^T)$. Therefore:

$$A_t^q x \sim N(\sqrt{\alpha_t} A_t^q (g \circ x_0), \sigma_t^2 A_t^q (A_t^q)^T)$$

111 The optimal \hat{f}^q satisfies the first-order condition:

$$0 = \mathbb{E}_{\substack{x_0 \sim X \\ g \sim T(2)}} \int p(A_t^q x | g \circ x_0) \left(\hat{f}^q(A_t^q x) - (g \circ x_0)^q \right) d(A_t^q x)$$

112 where the conditional density is:

$$p(A_t^q x | g \circ x_0) \propto \exp \left(-\frac{1}{2} \|A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0)\|_{\Sigma_q^{-1}}^2 \right)$$

113 Solving for \hat{f}^q yields equation (8). □

114 **Proposition A.10** (Diagonal operators and mask binarization). *When $A_t^q = \text{diag}(a_i^q)$ is a diagonal*
 115 *matrix with entries a_i^q , the optimal denoiser simplifies to:*

$$\hat{f}^q(x, t) = \sum_{i=1}^N \sum_{g \in T(2)} (g \circ x_0^i)^q \cdot \text{softmax}_{i,g} \left\{ -\frac{1}{2\sigma_t^2} \|B^q \odot (x - \sqrt{\alpha_t}(g \circ x_0^i))\|^2 \right\} \quad (9)$$

116 where B^q is the binary mask with $B_i^q = 1$ if $a_i^q > 0$ and $B_i^q = 0$ otherwise, and \odot denotes
 117 element-wise multiplication.

118 *Proof.* We begin with the generalized optimal denoiser from equation (8) and substitute the diagonal
 119 operator $A_t^q = \text{diag}(a_1^q, a_2^q, \dots, a_d^q)$.

120 For a diagonal matrix $A_t^q = \text{diag}(a_i^q)$, the covariance matrix becomes:

$$\Sigma_q = \sigma_t^2 A_t^q (A_t^q)^T = \sigma_t^2 \text{diag}([a_i^q]^2)$$

121 The Mahalanobis distance in equation (8) can then be written as:

$$\|A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i)\|_{\Sigma_q^{-1}}^2 = (A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i))^T \Sigma_q^{-1} (A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i))$$

122 Since A_t^q is diagonal, we have:

$$A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i) = \text{diag}(a_j^q) \cdot (x - \sqrt{\alpha_t}(g \circ x_0^i)) = (a_j^q (x_j - \sqrt{\alpha_t}(g \circ x_0^i)_j))_{j=1}^d$$

123 For any coordinates where $a_i^q = 0$ we can apply the same reasoning as in eq. (6) and just exclude
 124 them, by effectively projecting out. In the projected subspace of $a_i^q \neq 0$, the covariance matrix is
 125 now invertible: $\Sigma_q^{-1} = \text{diag}([a_j^q \sigma_t^2]^{-2})$ for $a_j^q \neq 0$. Therefore:

$$\begin{aligned} \|A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i)\|_{\Sigma_q^{-1}}^2 &= \sum_{j: a_j^q \neq 0} \frac{(a_j^q)^2 (x_j - \sqrt{\alpha_t}(g \circ x_0^i)_j)^2}{\sigma_t^2 (a_j^q)^2} \\ &= \frac{1}{\sigma_t^2} \sum_{j: a_j^q \neq 0} (x_j - \sqrt{\alpha_t}(g \circ x_0^i)_j)^2 \end{aligned}$$

126 **Key observation:** The coefficients a_j^q cancel out completely when $a_j^q \neq 0$. The actual values of
 127 non-zero a_j^q do not affect the optimal denoiser—only whether $a_j^q = 0$ or $a_j^q \neq 0$ matters.

128 We can now define the binary mask B^q with entries:

$$B_j^q = \begin{cases} 1 & \text{if } a_j^q \neq 0 \\ 0 & \text{if } a_j^q = 0 \end{cases}$$

129 This allows us to rewrite the distance as:

$$\|A_t^q x - \sqrt{\alpha_t} A_t^q (g \circ x_0^i)\|_{\Sigma_q^{-1}}^2 = \frac{1}{\sigma_t^2} \|B^q \odot (x - \sqrt{\alpha_t} (g \circ x_0^i))\|^2$$

130 where \odot denotes element-wise multiplication.

131 **Connection to patch-based denoiser:** When $a_j^q = 0$, the corresponding pixel j is effectively
 132 removed from the optimization, as it contributes zero to the distance metric. This is precisely the
 133 locality constraint from the patch-based denoiser: pixels outside the patch (where $a_j^q = 0$) do not
 134 influence the denoising of pixel q .

135 Substituting back into equation (8), we obtain:

$$\hat{f}^q(x, t) = \sum_{i=1}^N \sum_{g \in T(2)} (g \circ x_0^i)^q \cdot \text{softmax}_{i,g} \left\{ -\frac{1}{2\sigma_t^2} \|B^q \odot (x - \sqrt{\alpha_t} (g \circ x_0^i))\|^2 \right\}$$

136 This completes the proof, showing that the optimal denoiser depends only on the binary support of
 137 the diagonal operator, not on the specific non-zero values. \square

138 **Remark A.11** (Justification for binary masks). *The key insight is that when the masking operator*
 139 *A_t^q has a diagonal structure, the specific values of the non-zero entries cancel out in the softmax*
 140 *computation. This means that:*

- 141 1. *The optimal denoiser depends only on the pixels that are included in the mask (the support),*
 142 *not their relative weights.*
- 143 2. *Binary masks $\{0, 1\}$ are as expressive as any diagonal weighting scheme for this optimization*
 144 *problem.*
- 145 3. *This theoretical result justifies our practical choice of binary masks in the main paper, as*
 146 *more complex weighting provides no additional benefit for the optimal denoiser.*

147 A.4 “Pass-through” denoisers: detailed analysis of SNR

148 In this section, we provide a detailed analysis of the signal-to-noise ratio in the principal components
 149 of the data, extending section “Pass-through” denoisers in the main paper. Let’s consider the data
 150 matrix $X = [x_0^1 x_0^2 \dots x_0^N] \in \mathbb{R}^{d \times N}$. Doing singular value decomposition, and assuming $N \geq d$
 151 we get $X = U \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d) V^T$, where λ_i are sorted in the descending order of their absolute
 152 values. Covariance of the dataset, assuming that the mean of the dataset is zero:

$$\Sigma = \frac{1}{N} X X^T = \frac{1}{N} U \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_d^2) U^T,$$

153 where U are the principal components of the data and λ_i^2/N is the variance of the data along those
 154 components. We can now compute the signal-to-noise ratio along each of the principal components
 155 of the data:

$$\begin{aligned} \text{SNR}_i &= \frac{\mathbb{E}_{x_0 \sim X} \left[(U_i^T \sqrt{\alpha_t} x_0)^2 \right]}{\mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \left[(U_i^T \sqrt{1 - \alpha_t} \epsilon)^2 \right]} \\ &= \frac{\alpha_t \cdot U_i^T \Sigma U_i}{(1 - \alpha_t) \cdot U_i^T U_i} \\ &= \frac{\alpha_t \cdot \lambda_i^2 / N}{1 - \alpha_t} \end{aligned}$$

156 When $\lambda_i^2/N \gg (1 - \alpha_t)/\alpha_t$, the intrinsic data variance is much larger than the relative noise level,
 157 and the signal was not “destroyed” by noise. Note, the analysis above does not have to be performed

on the entire dataset, but rather on the most relevant set of neighbors to the image that is currently being denoised. In that case, the high SNR projections will be more precise and specific to each particular image as long as SVD is well defined. Due to computation constraints and to keep the analysis simple from now on we will assume that the covariance matrix is computed on the entire dataset.

A.5 Manipulating the sensitivity field: variance of the perturbation

In this section, we provide the derivation for the variance of the added perturbation λ_W in the section “Manipulating the sensitivity field” of the main paper. Denote by $v = \gamma cs$ the signal vector; then the empirical covariance of the modified data is

$$\begin{aligned}\Sigma_{\text{mod}} &= \mathbb{E}[\hat{x}_0 \hat{x}_0^T] \\ &= \mathbb{E}[(x_0 + \gamma cs)(x_0 + \gamma cs)^T] \\ &= \mathbb{E}[x_0 x_0^T] + \gamma \mathbb{E}[x_0 s^T c^T] + \gamma \mathbb{E}[cs^T x_0^T] + \gamma^2 \mathbb{E}[css^T c^T] \\ &= \Sigma_{\text{orig}} + \gamma^2 \mathbb{E}[cc^T] ss^T \\ &= \Sigma_{\text{orig}} + \gamma^2 \frac{1}{3} I_3 \otimes ss^T,\end{aligned}$$

where we used $\mathbb{E}[x_0] = 0$, $\mathbb{E}[c] = 0$, and for $c \sim \text{Uniform}([-1, 1]^3)$, we have $\mathbb{E}[cc^T] = \frac{1}{3} I$. For the noisy observations $\hat{x}_t = \sqrt{\alpha_t} \hat{x}_0 + \sqrt{1 - \alpha_t} \epsilon$, the covariance becomes:

$$\begin{aligned}\Sigma_{\text{mod}}^t &= \mathbb{E}[\hat{x}_t \hat{x}_t^T] \\ &= \alpha_t \mathbb{E}[\hat{x}_0 \hat{x}_0^T] + (1 - \alpha_t) I \\ &= \alpha_t \Sigma_{\text{mod}} + (1 - \alpha_t) I \\ &= \alpha_t \Sigma_{\text{orig}} + \alpha_t \gamma^2 \frac{1}{3} I_3 \otimes ss^T + (1 - \alpha_t) I.\end{aligned}$$

Assuming the RGB perturbation affects each color channel independently and focusing on a single channel, the second term contributes a rank-1 perturbation with eigenvalue $\lambda_W^2 = \alpha_t \gamma^2 \|s\|^2 / 3$.

By the Wiener filter analysis of ??, the learned sensitivity along the new "W" principal component is

$$\begin{aligned}s_w(t) &= \frac{\lambda_W^2}{\lambda_W^2 + (1 - \alpha_t)} \\ &= \frac{\alpha_t \gamma^2 \|s\|^2 / 3}{\alpha_t \gamma^2 \|s\|^2 / 3 + (1 - \alpha_t)} \\ &= \frac{\alpha_t \gamma^2 \|s\|^2}{\alpha_t \gamma^2 \|s\|^2 + 3(1 - \alpha_t)},\end{aligned}$$

where $(1 - \alpha_t)$ is the noise variance at timestep t .

B Additional Experiments and Ablation

B.1 Ablation of our model

The analytical model proposed in this paper has a single hyperparameter: th – the threshold of the sensitivity field binarization. In appendix A.3 we formally justify binarization of the sensitivity fields for our analytical model. Here we demonstrate the effect of choosing different binarization thresholds. In particular, from fig. 1 we can see that higher threshold values (i.e., smaller patch sizes) correspond to a sharper, but “patchier”. On the other side, small threshold values (i.e., bigger patch sizes) cause the generated image to be over-smoothed. We report the r^2 and MSE metrics of correlation with the trained diffusion model for different threshold values in table 1.

B.2 Self-attention layers in denoising U-Nets

Across our experiments, we are using a trained DDPM model with removed self-attention (SA) layers following [2]. In this section we demonstrate that removing the self-attention layer brings the FID

Table 1: Comparison of r^2 and MSE metrics across datasets for different binarization threshold values. Best values are highlighted in bold.

Threshold	CIFAR10		CelebA-HQ		MNIST		Fashion MNIST	
	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow
0.005	0.396	0.059	0.786	0.038	0.492	0.151	0.563	0.115
0.010	0.520	0.046	0.865	0.023	0.441	0.165	0.517	0.122
0.020	0.672	0.031	0.897	0.017	0.418	0.176	0.406	0.144
0.050	0.773	0.021	0.894	0.017	0.214	0.255	0.072	0.211
0.070	0.771	0.022	0.879	0.020	0.214	0.255	-0.192	0.264
0.100	0.737	0.026	0.852	0.024	0.214	0.255	-0.407	0.311
0.150	0.641	0.036	0.799	0.033	0.214	0.255	-0.209	0.270

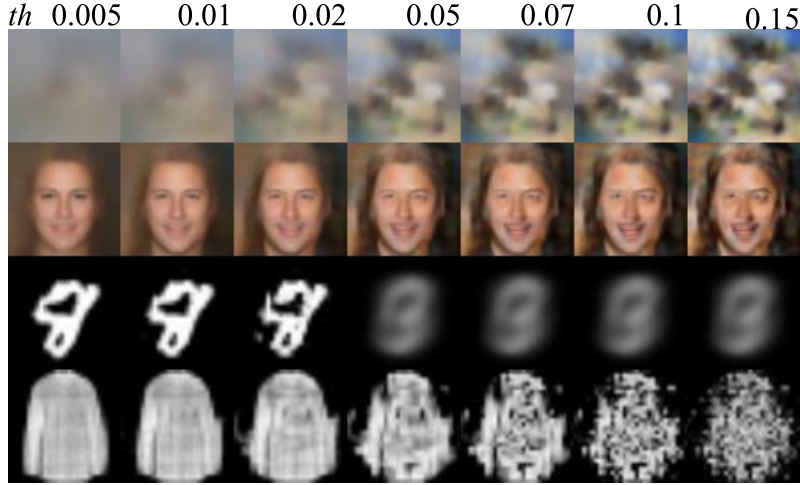


Figure 1: Ablation of the binarization threshold th .

score to 6.04 from 4.12 with SA. Qualitatively, the generated images look similar with and without SA and thus our analysis in the main paper can be extended to U-Nets with SA layers.

In particular, we train a U-Net without self-attention and compare it with a baseline U-Net trained with self-attention. Using the gradient-estimation sampler from [4], we report the FID scores for both models, and in Figure 2, we compare sample results.

B.3 How to reproduce the reported sensitivity fields

In this section, we provide the technical details and intuition needed to measure the sensitivity fields of diffusion models. All the results are reported for CIFAR10 dataset. Recall that the optimization problem is invariant under the change of variables from the initial image x_0 to the noise sample ϵ ; see appendix A.1 for details. Consequently, one can measure the sensitivity field in either the noise parameterization, $\partial\epsilon(x, t)/\partial x$, or the image parameterization, $\partial x_0(x, t)/\partial x$. Although the choice is merely a theoretical convenience, in practice the model’s behavior is highly sensitive to it.

The *top* row of fig. 3 shows the sensitivity fields of a DDPM model trained to predict x_0 and then reparameterized with a linear transform to predict ϵ ; here we plot $\partial\epsilon(x, t)/\partial x$. The *middle* row depicts the same model, but the sensitivity is evaluated in the image parameterization, i.e. $\partial x_0(x, t)/\partial x$. As we can see, a simple linear reparameterization applied to the model output drastically alters the result. These observations are intuitive. From the optimal-denoiser analysis, we know that, in the high-noise regime, the model predicts an image close to the dataset mean. Thus, predicting the added noise sample ϵ for each pixel q is almost equivalent to outputting q minus that mean, so the noise-parameterized sensitivity field appears highly local. Because this visualization is not very informative, we chose to plot $\partial x_0(x, t)/\partial x$ throughout the paper, as it captures the actual structure of the sensitivity field.



Figure 2: Samples from trained DDPM U-Nets without (left) and with (right) self-attention layers. The initial random noise is the same for both sets of images.

Figure 4 illustrates the effect of training U-Net and DiT models in the two parameterizations. Recall that $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon$. For large t where $\alpha_t \rightarrow 0$, image x_0 is ill-defined given ϵ and x_t ; conversely, for small t where $\alpha_t \rightarrow 1$, ϵ is ill-defined given x_0 and x_t . Hence, while theory predicts identical results (up to re-parameterization), numerical errors lead to different behavior at low and high noise levels. The top two rows of fig. 4 show $\partial x_0(x, t)/\partial x$ for models trained in the noise parameterization, revealing a pronounced shrinkage of the sensitivity fields in the high-noise regime. We hypothesize that this is a numerical artifact and therefore plot, in the bottom two rows, the fields obtained from models trained directly in the image parameterization. For clarity, all DDPM examples in the main paper are trained in that setting.

Finally, the *middle* and *bottom* rows of fig. 3 compare two normalization strategies. In the middle row, each sensitivity field is normalized independently to $[-1, 1]$; in the bottom row, the images are normalized jointly, preserving relative scale. Joint normalization makes the field appear less local while preserving its overall mass. Throughout the paper, we adopt per-image normalization, as it more faithfully reflects the binarization assumed in our analytical model.

Summary of visualization choices

- Train the model to predict the image x_0 (not the noise ϵ).
- Visualize the sensitivity of the image prediction, i.e. $\partial x_0(x, t)/\partial x$.
- Apply per-image normalization.

B.4 Sensitivity field of the optimal denoiser

In this section, we provide a visualization of the sensitivity fields of the optimal denoiser on the CIFAR10 dataset. As shown in fig. 5, the sensitivity of the optimal denoiser closely resembles that of the trained models only in the high-noise regime. At intermediate noise levels, the sensitivity field begins to diverge, and in the low-noise regime, it ultimately “explodes”.

B.5 Generation dynamics

Here we provide additional results demonstrating the dynamics of image generation. In fig. 7 we numerically compare x_0 predictions through the generation process.

In fig. 6 we demonstrate how the dynamics of image generation of our analytical model compares with that one of a trained DDPM model. Note that the trained model produces noisy single-step

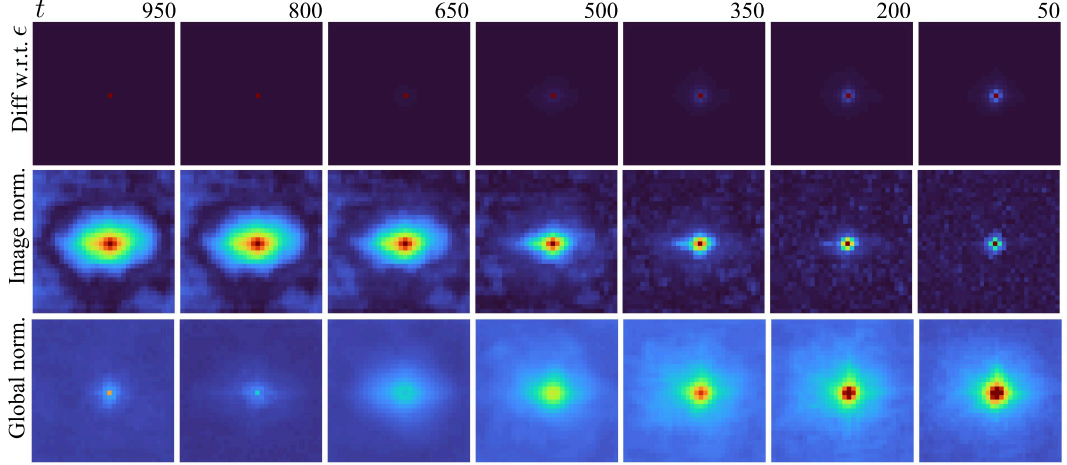


Figure 3: **Top:** sensitivity field of the noise prediction $\partial\epsilon(x, t)/\partial x$. **Middle:** sensitivity field of the image prediction $\partial x_0(x, t)/\partial x$ with per-image normalization. **Bottom:** the same field with joint normalization across images.

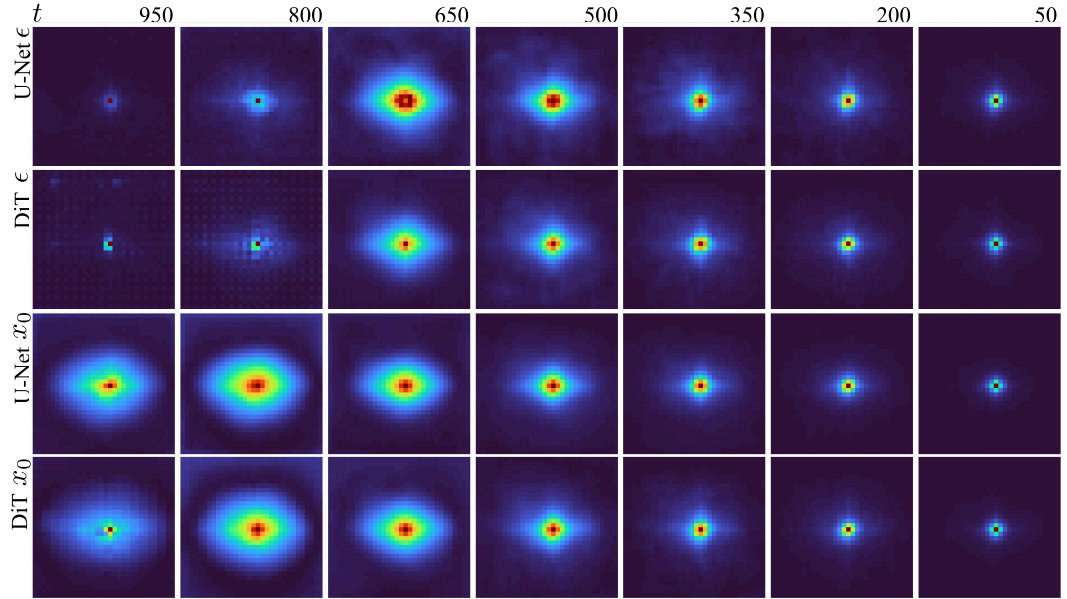


Figure 4: Sensitivity fields $\partial x_0(x, t)/\partial x$ for U-Net (left) and DiT (right). **Top two rows:** models trained to predict noise ϵ . **Bottom two rows:** models trained to predict the image x_0 . The shrinkage observed at high noise in the noise-parameterized models is likely due to numerical instability.

235 predictions for high noise levels ($t \geq 850$). We explain this behavior by the fact that the model was
 236 trained to predict ϵ and later re-parametrized to output x_0 for the visualization. Since $\alpha_t \rightarrow 0$ for
 237 high noise level, x_0 becomes ill-defined and thus noises the outputs.

238 B.6 Variance of the reported metrics

239 In table 2 we report standard deviation values for the metrics reported in Table 1 of the main paper.
 240 All the values are calculated across 128 samples.

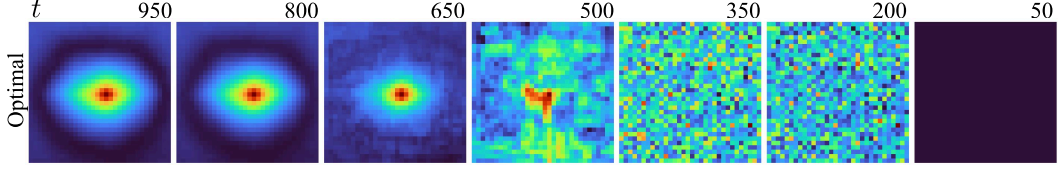


Figure 5: Sensitivity field of the optimal denoiser.

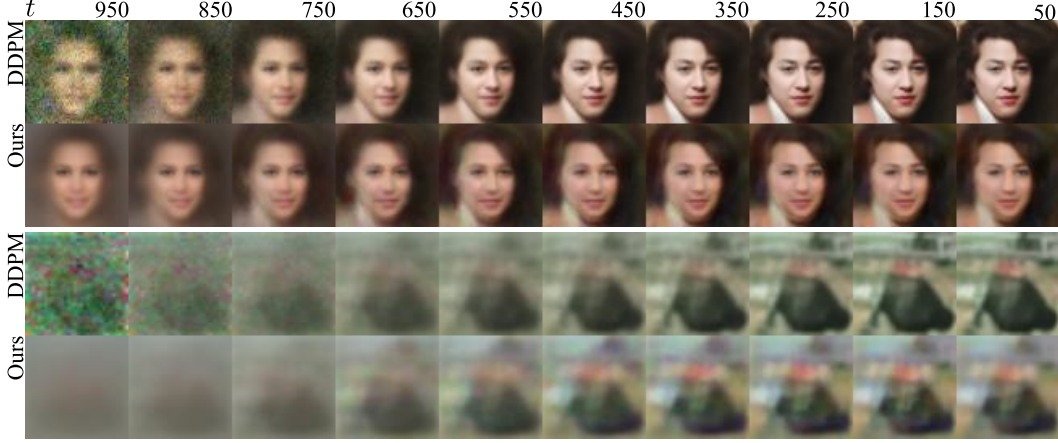


Figure 6: Intermediate generation results of a trained DDPM model (rows 1 and 3) and ours (rows 2 and 4). The figure displays single-step estimations of x_0 from each x_t along a sampling trajectory of 10 steps.

241 B.7 Quantitative measure of novelty of samples

242 In this work, we focus on the ability of the trained diffusion models to generate novel samples that
 243 contrast with the behavior of the optimal denoiser. Therefore, the ability of the analytical model to
 244 generate novel samples is paramount. In figure 5 of the main paper (as well as in appendix B.8) we
 245 report the nearest neighbors from the training dataset for each sample generated with our analytical
 246 model. To quantify these results, we report the average L_2 distances between samples generated with
 247 each of the baseline models and the closest image in the dataset in table 3. Additionally, we report
 248 the dynamics of the “novelty” measure in the generation process in fig. 8.

249 B.8 Additional generation results

250 We present additional generation results similar to fig. 5 of the main paper in figs. 9 to 13.

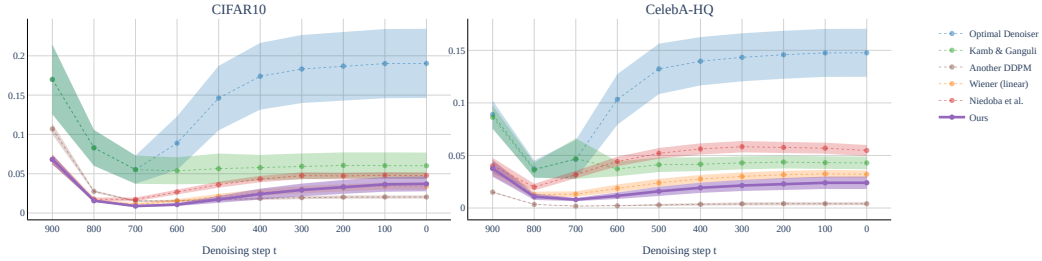


Figure 7: Mean Squared Error (MSE) between the baseline’s predictions and a trained DDPM model. The MSE is calculated on x_0 prediction from each x_t point along a 10-step generation trajectory. The results are presented on the CIFAR10 and CelebA-HQ datasets. Mean and standard deviation values were calculated across 128 samples.

Table 2: We report standard deviations values for the metrics reported in Table 1 of the main paper. All values are computed over 128 samples.

Method	CIFAR10		CelebA-HQ		AFHQv2		MNIST		Fashion MNIST	
	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow	$r^2 \uparrow$	MSE \downarrow
Optimal Denoiser	± 0.774	± 0.044	± 0.298	± 0.023	± 0.371	± 0.023	± 0.204	± 0.036	± 0.344	± 0.077
Wiener (linear)	± 0.092	± 0.004	± 0.039	± 0.004	± 0.072	± 0.003	± 0.066	± 0.014	± 0.068	± 0.018
Kamb & Ganguli [2]	± 0.126	± 0.017	± 0.081	± 0.006	± 0.081	± 0.006	± 0.110	± 0.045	± 0.114	± 0.032
Niedoba et al.[3]	± 0.137	± 0.004	± 0.092	± 0.005	± 0.092	± 0.006	± 0.077	± 0.022	± 0.088	± 0.022
Ours	± 0.078	± 0.008	± 0.032	± 0.006	± 0.026	± 0.004	± 0.051	± 0.015	± 0.042	± 0.011
Another DDPM	± 0.113	± 0.002	± 0.007	± 0.001	± 0.019	± 0.001	± 0.082	± 0.019	± 0.020	± 0.005

Table 3: We numerically quantify the ability of analytical models to produce images outside of the training dataset. In this table, we provide the average $L2$ distance between images generated with the baselines and the corresponding closest image in the training dataset. Results are averaged over 128 samples.

Method	CIFAR10	CelebA-HQ	AFHQv2	MNIST	Fashion MNIST
Optimal Denoiser	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Wiener (linear)	0.091 ± 0.006	0.104 ± 0.006	0.112 ± 0.005	0.177 ± 0.015	0.133 ± 0.013
Kamb & Ganguli [2]	0.094 ± 0.005	0.089 ± 0.006	0.136 ± 0.007	0.355 ± 0.061	0.218 ± 0.032
Niedoba et al.[3]	0.113 ± 0.007	0.134 ± 0.009	0.145 ± 0.007	0.221 ± 0.019	0.215 ± 0.019
Ours	0.040 ± 0.005	0.063 ± 0.004	0.063 ± 0.004	0.204 ± 0.023	0.131 ± 0.027
Another DDPM	0.079 ± 0.014	0.087 ± 0.010	0.095 ± 0.013	0.103 ± 0.023	0.067 ± 0.015

C Implementation details

C.1 Sampling.

In all of the generations in this paper, we are using diffusers’ [1] implementation of the DDIM [5] sampler with 10 sampling steps. We discretize the noise time scale for $t \in [1, 1000]$ where $t = 0$ is no noise and $t = 1000$ is full noise. The scheduler is linear with $\alpha_0 = 10^{-4}$ and $\alpha_{1000} = 0.02$.

C.2 Training DDPM Model

We train a Denoising Diffusion Probabilistic Model (DDPM) U-Net using a third-party pytorch implementation [6]. We adopt the U-Net model architecture based on the input image resolution:

- *MNIST/FashionMNIST* (`img_size` = 28): 3 downsampling levels with `channel_mult` = [1, 2, 2], base channel width 64.
- *CIFAR10* (`img_size` = 32) and *CelebA-HQ/AFHQ* (`img_size` = 64): 4 downsampling levels with `channel_mult` = [1, 2, 3, 4], base channel width 128.

The number of residual blocks per level is fixed to 2, with no self-attention modules included. Dropout is set to 0.15 throughout the network. The model is trained for 200 epochs with a batch size of 32. We use the Adam optimizer with a learning rate of 10^{-4} over 1000 diffusion steps. Training and evaluation use fixed random seeds for reproducibility.

C.3 Our analytical model

Below we provide a detailed description of the implementation of our analytical model. A key component of this implementation is the weighted streaming softmax (*wssm*) that accumulates the product $x_0 \text{softmax}(\dots)$ over batches of training images. Additionally we attach the code of our algorithm, the weighted streaming softmax and of all the baselines to the supplementary materials.

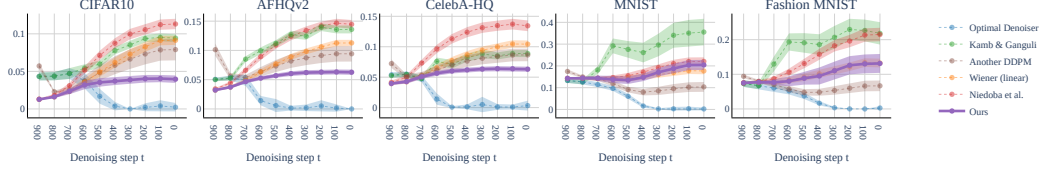


Figure 8: L_2 distance between x_0 prediction and the closest image in the training dataset reported along a 10-step generation trajectory for 5 datasets.

Algorithm 1 Single denoising step of the proposed analytical model.

Require: Noisy image x_t
Timestep t
Precomputed covariance S of the data
Mask threshold th
Dataset X
Schedule of α_t

Ensure: Estimated clean image \hat{x}_0

- 1: $U\Lambda U^\top = S$ ▷ SVD of the covariance matrix
- 2: $W_t = U \text{diag} \left(\frac{\alpha_t \lambda_i}{\beta_t + \alpha_t \lambda_i} \right) U^\top$ ▷ Current Wiener matrix
- 3: $M_t = \text{Binarize}(W_t, th)$ ▷ Construct the projection matrix
- 4: $wssm.init()$ ▷ Initialize weighted streaming softmax
- 5: **for** each batch $x_0^{(k)}$ from X **do**
- 6: $D_k = \text{stack} \left[\left(x_t - \sqrt{\alpha_t} x_0^{(k)} \right)^2 \right]$ ▷ Distance to x_t for each image in the batch
- 7: $Dm_t = D_k M_t$ ▷ Each row of M_t serves as a mask
- 8: $wssm.update \left(-Dm_t/2 [1 - \alpha_t], x_0^{(k)} \right)$ ▷ Add the distances and the value
- 9: **end for**
- 10: $\hat{x}_0 = wssm.value()$
- 11: **return** \hat{x}_0

272 C.4 Baseline implementation details

273 **Wiener filter.** To implement the Wiener matrix, we first center each dataset to zero mean. Then
274 we pre-compute the covariance matrix of the dataset. Note that this is part of “training” and these
275 computations were not included in the runtime report. On sampling, use the PyTorch implementation
276 of SVD to compute the principal components and the corresponding singular values. Finally, we are
277 using eq. (7) from the main paper to implement W_t . Note that we are using the Wiener filter as a
278 denoiser, and when generating the images, we are still using a 10-step DDIM sampling, effectively
279 applying the Wiener filter 10 times to the initial noise.

280 **Kamb & Ganguli model.** We implemented the analytical model suggested by Kamb & Ganguli in
281 our code base. Then we fit the patch sizes M_t of the analytical model to our trained DDPM U-Nets,
282 maximizing the r^2 between the scores on each step of generation. Below are the patch sizes that we
283 obtained:

- 284 • **CIFAR10** 32×32 : [32, 32, 32, 29, 25, 17, 13, 9, 7, 3]
- 285 • **CelebA-HQ** 64×64 : [64, 64, 57, 49, 45, 25, 17, 17, 9, 3]
- 286 • **AFHQ** 64×64 : [64, 64, 45, 33, 25, 17, 17, 9, 9, 3]
- 287 • **MNIST** 28×28 : [28, 28, 23, 23, 17, 17, 17, 13, 9, 9]
- 288 • **Fashion MNIST** 28×28 : [28, 28, 25, 23, 17, 17, 13, 13, 9, 9]

289 **Niedoba et al. model.** We similarly re-implement the Niedoba et al. model in our code base. We
290 then measure and average the sensitivity field of our instance of the DDPM model. The sensitivity

field is averaged across all output pixels and 64 different noise samples. After the averaging, we binarize the sensitivity field with a $th = 0.05$ relative to the max value in the receptive field for each timestep t . We empirically observed that the performance is worse with equivariance, so our implementation does not contain equivariance to translations. Finally, we implemented the algorithm similar to our model, but with the measured sensitivity fields.

C.5 Computational resources and runtime

All the experiments were performed on a server machine with *Ubuntu 20.04*. The machine has *1008GB* RAM, *128* CPU cores and $8 \times$ *NVIDIA RTX A6000* GPUs with *49140MB* VRAM. We note that all the baselines could be run with fewer computational resources. In table 4 we provide the average run times for each baseline.

Table 4: We demonstrate the computational efficiency of each method by displaying the total sampling time for each of the baselines over 10 denoising steps. None of the methods are optimized for runtime, and the comparison is provided only as a rough reference. Results show times averaged over 64 samples.

Method	CIFAR10	CelebA-HQ	AFHQv2	MNIST	Fashion MNIST
Optimal Denoiser	7.90	18.90	10.01	0.63	0.64
Wiener (linear)	0.11	3.10	3.08	0.07	0.07
Kamb & Ganguli [2]	44.44	349.68	181.08	4.40	4.49
Niedoba et al.[3]	22.35	76.09	322.65	22.77	23.19
Ours	21.25	70.23	314.55	22.39	22.97
Another DDPM	0.57	0.65	0.65	0.61	0.63

300



Figure 9: Additional generation results for all baselines and ours on the CelebA-HQ dataset.



Figure 10: Additional generation results for all baselines and ours on the AFHQ dataset.



Figure 11: Additional generation results for all baselines and ours on the CIFAR10 dataset.

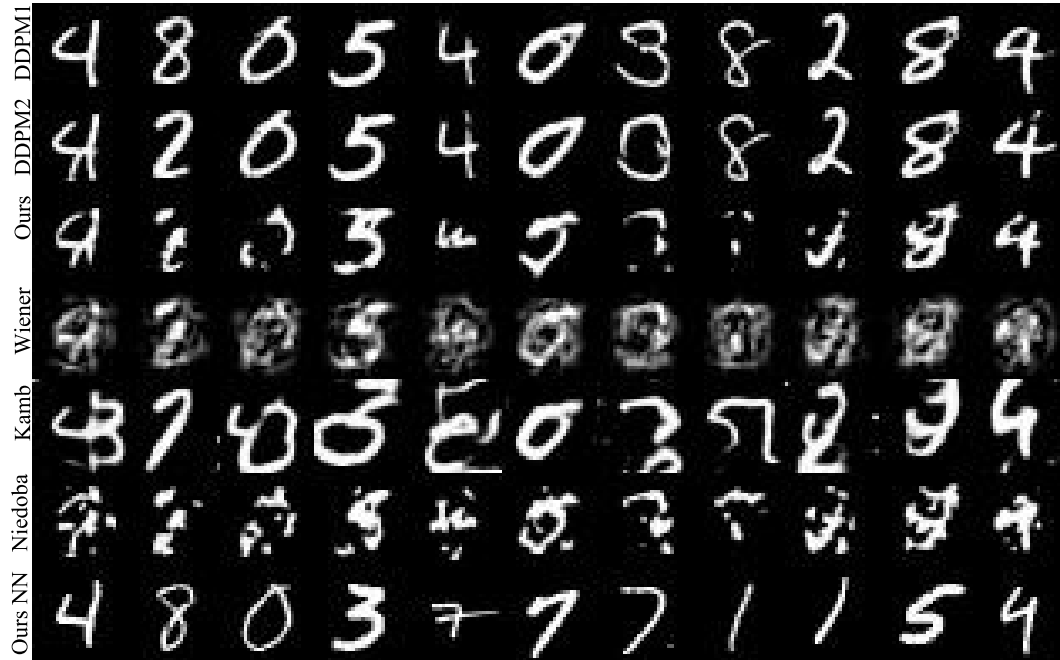


Figure 12: Additional generation results for all baselines and ours on the MNIST dataset.

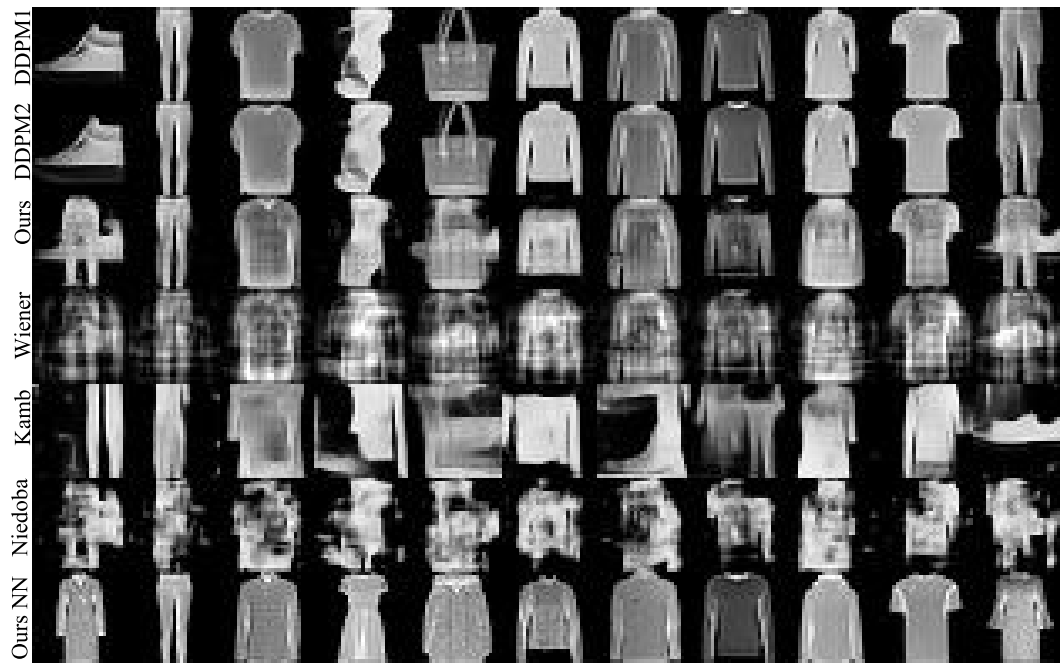


Figure 13: Additional generation results for all baselines and ours on the Fashion MNIST dataset.

References

- [1] Hugging Face. Diffusers library documentation. <https://huggingface.co/docs/diffusers/en/index>, 2024. Accessed: 2025-05-23.
- [2] Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.
- [3] Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. *arXiv preprint arXiv:2411.19339*, 2024.
- [4] Frank Permenter and Chenyang Yuan. Interpreting and improving diffusion models from an optimization perspective. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 40461–40483. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/permenter24a.html>.
- [5] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [6] Bohao Zou. Denoising diffusion probabilistic model (ddpm) implementation. <https://github.com/zoubohao/DenoisingDiffusionProbabilityModel-ddpm>, 2022. Accessed: 2025-05-23.