From analytic CNN surrogates to recovery guarantees for learned inverse-problem solvers

Minh-Hai NGUYEN

3rd year PhD student at IRIT & CBI, Université de Toulouse

Joint work with Quoc-Bao DO, Edouard PAUWELS and Pierre WEISS

SO Seminar, IMT, 17 November 2026.







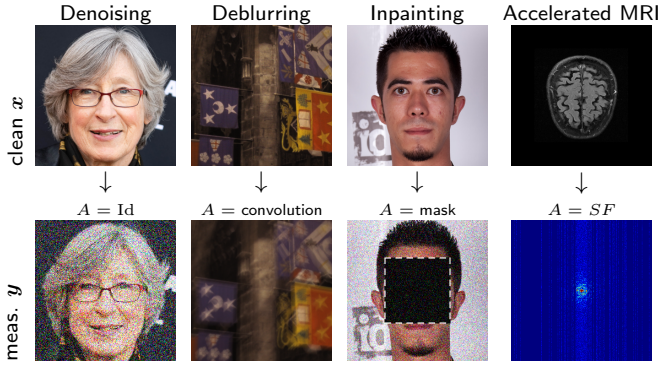
Inverse problems

Forward model

Let $A : \mathbb{R}^N \to \mathbb{R}^M$ be a linear operator, consider the acquisition model:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{e}, \qquad \text{where } \boldsymbol{e} \sim \mathcal{N}\left(0, \sigma^2 \mathrm{Id}\right)$$

Inverse problems: recover the clean image $\boldsymbol{x}$ from the measurements $\boldsymbol{y}$ and the forward operator A.



Inverse problems are ill-posed: a prior is needed

Imaging inverse problem are typically ill-posed: a solution may not exist, may not be unique (e.g, $\ker A \neq \{0\}$).

$\Rightarrow$ Example: the naive inversion $A^+\boldsymbol{y}$ amplifies the noise: we need a **prior** on $\boldsymbol{x}$.



clean $\boldsymbol{x}$ | $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{e}$ 24.4 dB | $A^+\boldsymbol{y}$ 6.8 dB

deblurring with 5% noise: the inversion is worse than the measurement

Variational regularization

$$\hat{x} \in \operatorname*{argmin}_x \ \tfrac{1}{2}\|Ax - y\|^2 + \lambda \mathcal{R}(x)$$

hand-crafted prior $\mathcal{R}$: Tikhonov, total variation, sparsity, ...

Learning-based methods

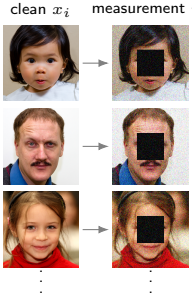
the prior is **learned from data**:

- plug-and-play priors, unrolled networks, generative (diffusion) priors, ...
- **supervised learning**: a network trained on pairs of clean images and measurements.

Today: the supervised learning framework.

Supervised reconstruction

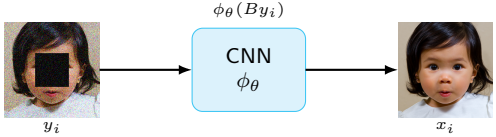
Clean-image dataset and measurements



$\mathcal{D} = \{x_1, \dots, x_D\}$: clean images

e.g. FFHQ or ImageNet; $y_i = Ax_i + e_i$

One network mapping measurements to clean images



trained so that $\phi_\theta(By_i) \approx x_i$ on the whole dataset

B is a back-projection operator (e.g., $B = A^T$ or $B = A^+$)



blurred y **deblurring** By k-space y **MRI** By

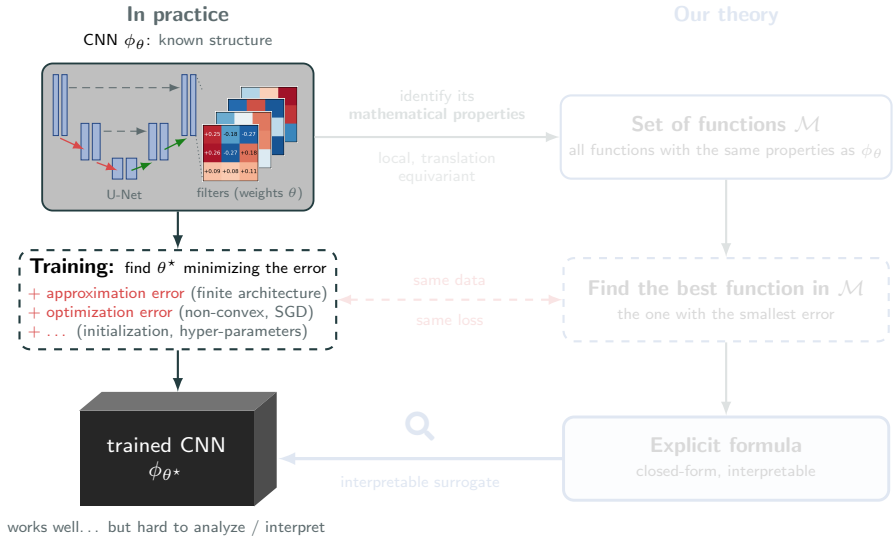
B brings the measurement back to image space, before the network sees it

Training objective (empirical risk minimization)

$$\theta^\star \in \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{D} \sum_{i=1}^{D} \|\phi_\theta(By_i) - x_i\|^2, \qquad \text{then} \quad \hat{x}(y) = \phi_{\theta^\star}(By).$$

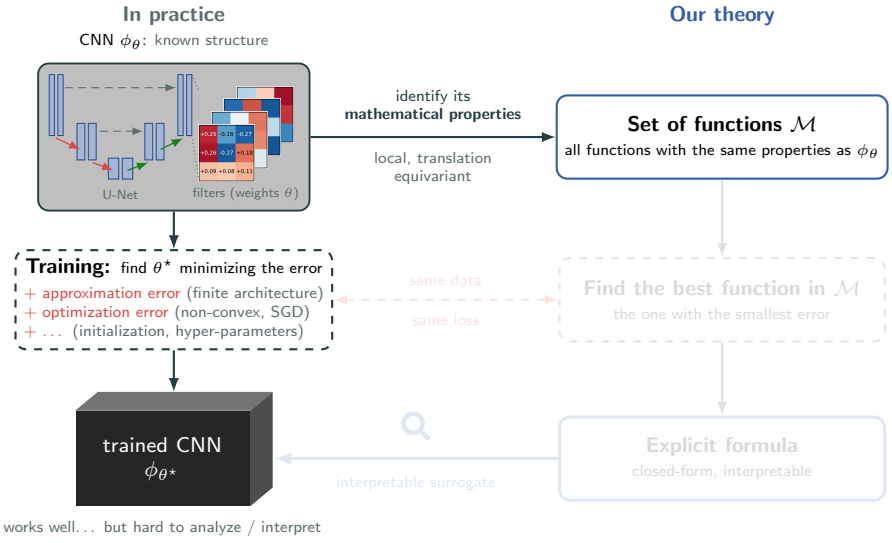
What does this trained network converge to?

A surrogate analytical model to understand neural networks



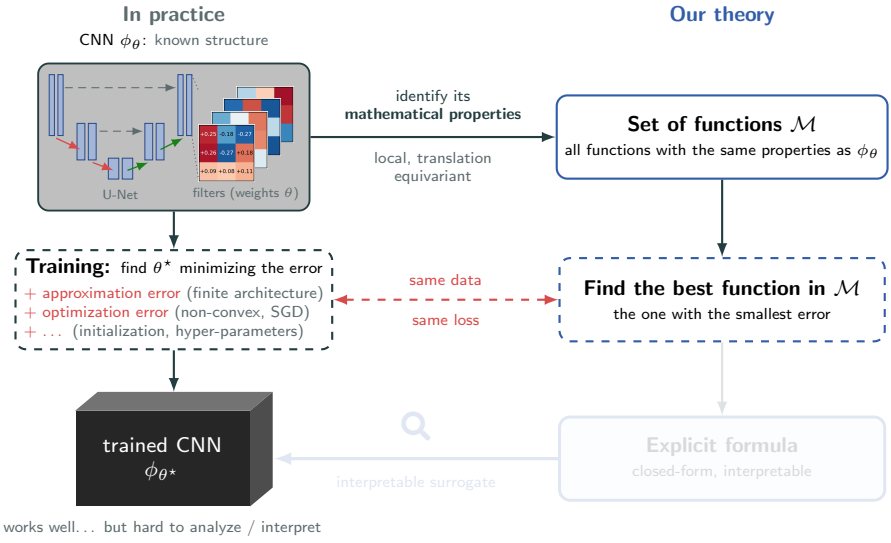
The formula **predicts and explains** what the trained CNN computes.

A surrogate analytical model to understand neural networks



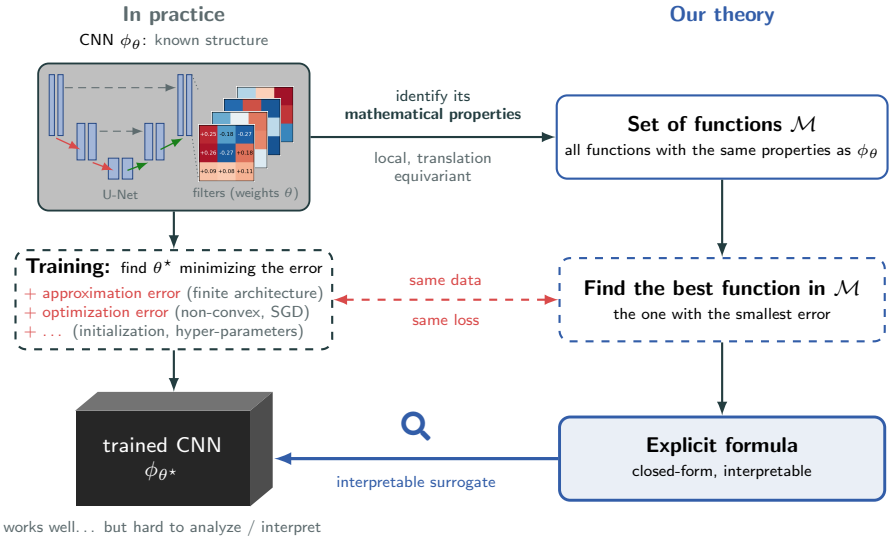
The formula **predicts and explains** what the trained CNN computes.

A surrogate analytical model to understand neural networks



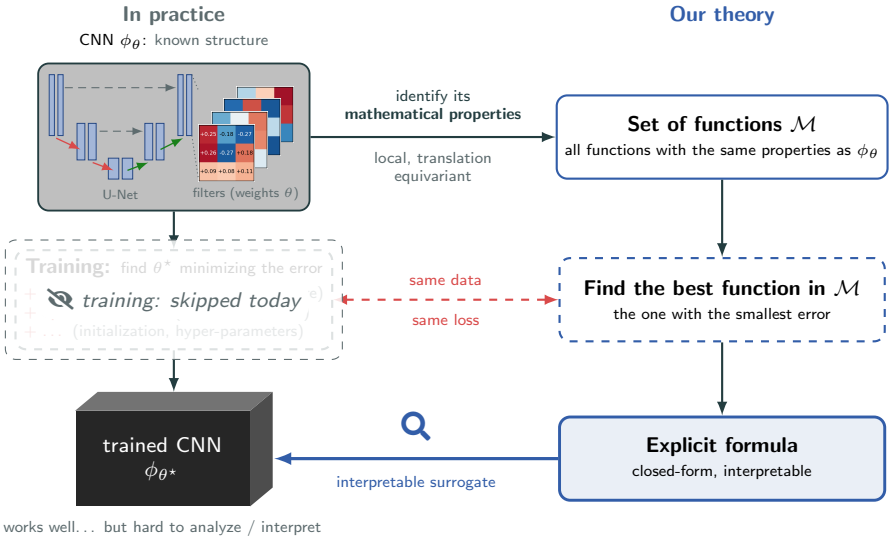
The formula **predicts and explains** what the trained CNN computes.

A surrogate analytical model to understand neural networks



The formula **predicts and explains** what the trained CNN computes.

A surrogate analytical model to understand neural networks



Today: ① the **optimal theoretical solution** ② it matches the output of **trained CNNs**

MSE training learns a conditional mean

$\mathcal{M}$ = **all measurable functions** (no architectural constraint)

Supervised learning

$$\theta^\star \in \underset{\theta\in\Theta}{\operatorname{argmin}}\ \mathbb{E}\left[\|\phi_\theta(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right]$$

Ideal target: the best function in $\mathcal{M}$

$$\phi^\star \in \underset{\phi\in\mathcal{M}}{\operatorname{argmin}}\ \mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right]$$

then $\phi^\star(z) = \mathbb{E}\left[\boldsymbol{x}|B\boldsymbol{y} = z\right]$, the **(empirical) MMSE estimator**.

If the neural network is **sufficiently expressive** and **well-trained**, $\phi_{\theta^\star}$ *approximates* the empirical MMSE estimator:

$$\boxed{\phi_{\theta^\star} \approx \phi^\star}$$

Link to generative models

Denoising ($A = B = \mathrm{Id}$):

$$D_\sigma(y) = \mathbb{E}\left[\boldsymbol{x}|\boldsymbol{x} + \sigma\varepsilon = y\right]$$

For Gaussian denoising, this same conditional mean determines the diffusion score and the flow-matching velocity.

MSE training learns a conditional mean

$\mathcal{M}$ = **all measurable functions** (no architectural constraint)

Supervised learning

$$\theta^\star \in \underset{\theta\in\Theta}{\operatorname{argmin}}\ \mathbb{E}\left[\|\phi_\theta(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right]$$

Ideal target: the best function in $\mathcal{M}$

$$\phi^\star \in \underset{\phi\in\mathcal{M}}{\operatorname{argmin}}\ \mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right]$$

then $\phi^\star(z) = \mathbb{E}\left[\boldsymbol{x}|B\boldsymbol{y} = z\right]$, the **(empirical) MMSE estimator**.

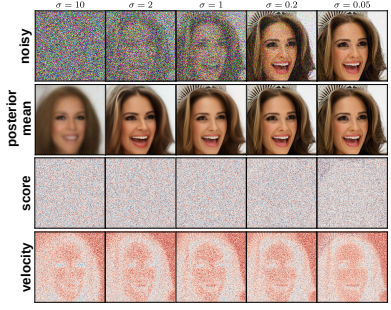
If the neural network is **sufficiently expressive** and **well-trained**, $\phi_{\theta^\star}$ *approximates* the empirical MMSE estimator:

$$\boxed{\phi_{\theta^\star} \approx \phi^\star}$$

Link to generative models

Denoising ($A = B = \mathrm{Id}$):

$$D_\sigma(y) = \mathbb{E}\left[\boldsymbol{x}|\boldsymbol{x} + \sigma\boldsymbol{\varepsilon} = y\right]$$



For Gaussian denoising, this same conditional mean determines the diffusion score and the flow-matching velocity.

No architectural constraint: the MMSE memorizes

$\mathcal{M}$ = **all measurable functions** $\Rightarrow$ the best function in $\mathcal{M}$ is the **MMSE estimator** $\phi^\star$

The MMSE estimator writes, for any $y \in \mathbb{R}^M$,

$$\hat{x}_{\text{MMSE}}(y) = \phi^\star(By) = \sum_{x \in \mathcal{D}} x \cdot w(x|y), \qquad \text{where} \quad w(x|y) \propto \mathcal{N}\left(By; BAx, \sigma^2 BB^\top\right).$$

- The weights are non-negative and sum to 1.
- **Memorization**: the reconstruction $\hat{x}_{\text{MMSE}}(y)$ is in the **convex hull** of the training images.
- At low noise level $\sigma \to 0$: nearest neighbor search in the training set.

Although $\hat{x}_{\text{MMSE}}$ is optimal over all measurable maps, it did not reproduce practical CNN behavior in our experiments.

$\Rightarrow$ **Practical CNNs need a model of their architectural constraints.**

No architectural constraint: the MMSE memorizes

$\mathcal{M}$ = **all measurable functions** $\Rightarrow$ the best function in $\mathcal{M}$ is the **MMSE estimator** $\phi^\star$

The MMSE estimator writes, for any $y \in \mathbb{R}^M$,

$$\hat{x}_{\mathsf{MMSE}}(y) = \phi^\star(By) = \sum_{x \in \mathcal{D}} x \cdot w(x|y), \qquad \text{where} \quad w(x|y) \propto \mathcal{N}\left(By; BAx, \sigma^2 BB^\top\right).$$

- The weights are non-negative and sum to 1.
- **Memorization**: the reconstruction $\hat{x}_{\mathsf{MMSE}}(y)$ is in the **convex hull** of the training images.
- At low noise level $\sigma \to 0$: nearest neighbor search in the training set.

Although $\hat{x}_{\mathsf{MMSE}}$ is optimal over all measurable maps, it did not reproduce practical CNN behavior in our experiments.

$\Rightarrow$ **Practical CNNs need a model of their architectural constraints.**

Inductive biases of CNNs: locality and translation equivariance

Two inductive biases of CNNs

- **Translation equivariance**: the same rule acts at every location
- **Locality**: an output pixel sees only an input patch.

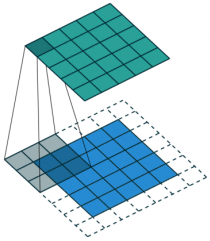
Define:

- $\mathcal{M}_{\mathsf{LE}}$: the set of local and translation equivariant functions
- $\mathcal{M}_\Theta$: the set of functions representable by a CNN with parameters $\theta \in \Theta$
- $\mathcal{M}$: the set of all measurable functions

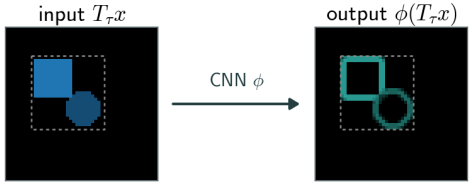
then,

$$\mathcal{M}_\Theta \subseteq \mathcal{M}_{\mathsf{LE}} \subseteq \mathcal{M}$$

What is the best estimator in $\mathcal{M}_{\mathsf{LE}}$? Can we compute it?



one *local* window, the *same* weights everywhere (circular padding)



$\phi(T_\tau x) = T_\tau\, \phi(x)$: the output shifts by the *same* τ as the input

Analytic formula: compares patches, averages pixels

Theorem (Closed-form of LE-MMSE)

Suppose that the N matrices $Q_n \overset{\text{def}}{=} \Pi_n B \in \mathbb{R}^{P \times M}$ have the same rank $r > 0$ for any n[1]. The LE-MMSE estimator $\hat{x}_{\text{LE-MMSE}}$ admits, for any $y \in \mathbb{R}^M$ the following expression, defined pixel-wise for each pixel n':

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{x \in \mathcal{D}} \sum_{n=1}^{N} x[n] \cdot w_{n',n}(x|y), \tag{1}$$

where the weights $w_{n',n}(x|y) \propto \mathcal{N}\left(Q_{n'}y; Q_n A x, \sigma^2 Q_n Q_n^\top\right)$.

Key properties

- The value at pixel n' of the LE-MMSE estimator is a weighted average of **all** the pixels in the training images. The reconstruction is a **patchwork** of training patches.
- It can produce images outside $\mathrm{conv}(\mathcal{D})$.

⚠ **Scope:** valid for a **class** of CNNs: **local and translation-equivariant** (finite receptive field, circular padding), MSE loss. **Not** for any CNN used in practice.

[1] This assumption can be relaxed: a general formula by rank stratification is also available.

Analytic formula: compares patches, averages pixels

Theorem (Closed-form of LE-MMSE)

Suppose that the N matrices $Q_n \overset{\text{def}}{=} \Pi_n B \in \mathbb{R}^{P \times M}$ have the same rank $r > 0$ for any n[1]. The LE-MMSE estimator $\hat{x}_{\text{LE-MMSE}}$ admits, for any $y \in \mathbb{R}^M$ the following expression, defined pixel-wise for each pixel n':

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{x \in \mathcal{D}} \sum_{n=1}^{N} x[n] \cdot w_{n',n}(x|y), \tag{1}$$

where the weights $w_{n',n}(x|y) \propto \mathcal{N}\left(Q_{n'}y; Q_n A x, \sigma^2 Q_n Q_n^\top\right)$.

Key properties

- The value at pixel n' of the LE-MMSE estimator is a weighted average of **all** the pixels in the training images. The reconstruction is a **patchwork** of training patches.
- It can produce images outside $\mathrm{conv}(\mathcal{D})$.

⚠ **Scope:** valid for a **class** of CNNs: **local and translation-equivariant** (finite receptive field, circular padding), MSE loss. **Not** for any CNN used in practice.

[1] This assumption can be relaxed: a general formula by rank stratification is also available.

The formula: compare patches, average centres

Look around pixel n'

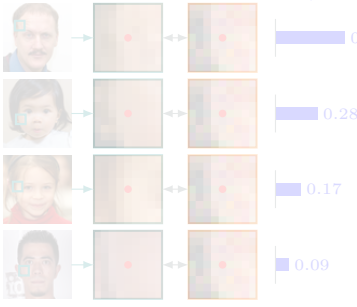
measurement By



patch $Q_{n'}y$

Compare with every patch of every training image

training image x | its patch $Q_n Ax$ | query patch $Q_{n'}y$ | weight $w_{n',n}(x|y)$



$\cdots$ millions of patches, most with $w \approx 0$

Average their centres

reconstruction $\hat{x}_{\text{LE-MMSE}}(y)$

$\sum_{x,\,n} w \times x[n]$

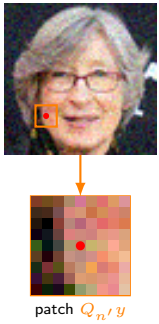
new value of n'
(repeat for every pixel)

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{\substack{\text{training} \\ \text{images } x \in \mathcal{D}}} \sum_{\substack{\text{positions} \\ n}} w_{n',n}(x|y)\, x[n], \qquad w_{n',n}(x|y) \propto \exp\Big(- \frac{\big\| Q_{n'}y - Q_n Ax \big\|^2_{Q_n}}{2\sigma^2} \Big)$$

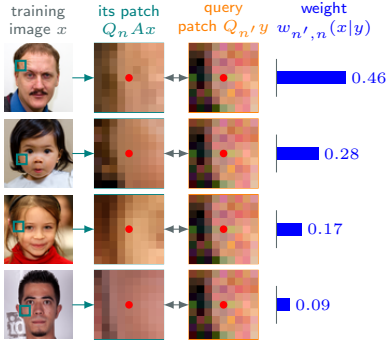
Similar patches $\Rightarrow$ large w. The weights add up to 1. Metric: $\|v\|_{Q_n} \stackrel{\text{def}}{=} \|Q_n^+ v\|$, with $Q_n = \Pi_n B$.

The formula: compare patches, average centres

Look around pixel n'



Compare with every patch of every training image

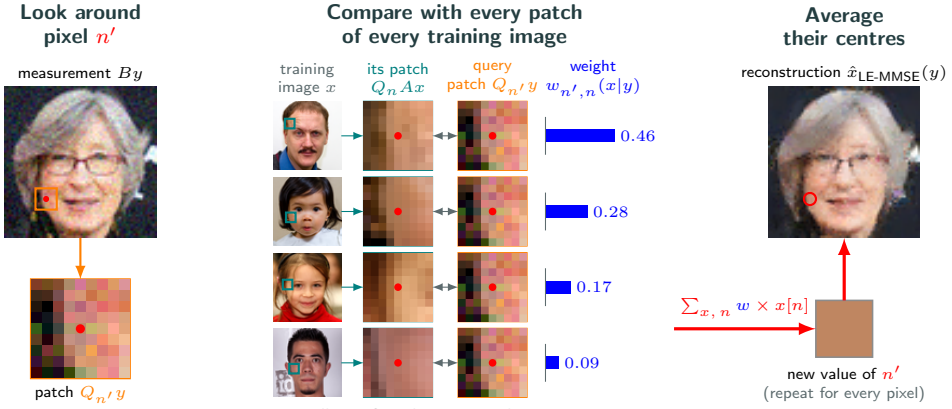


Average their centres

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{\substack{\text{training}\\ \text{images } x\in\mathcal{D}}} \sum_{\substack{\text{positions}\\ n}} w_{n',n}(x|y)\; x[n], \qquad w_{n',n}(x|y) \propto \exp\Big(-\frac{\big\|Q_{n'}y - Q_nAx\big\|^2_{Q_n}}{2\sigma^2}\Big)$$

Similar patches $\Rightarrow$ large w. The weights add up to 1. Metric: $\|v\|_{Q_n} \overset{\text{def}}{=} \|Q_n^+ v\|$, with $Q_n = \Pi_n B$.

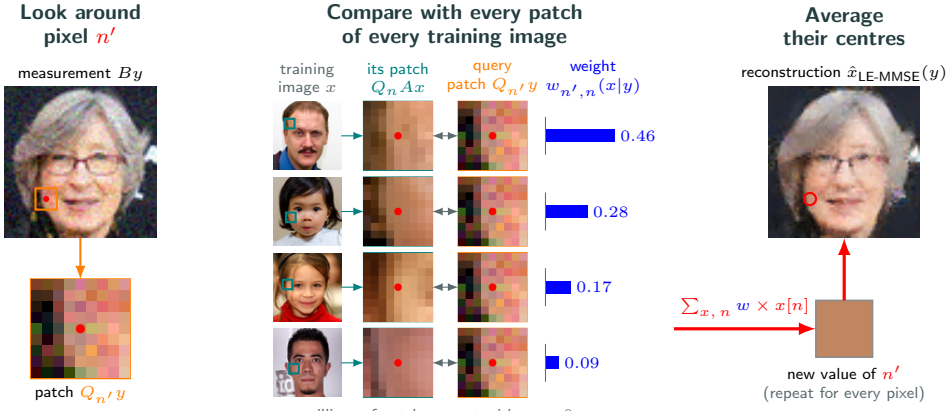
The formula: compare patches, average centres



$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{\substack{\text{training} \\ \text{images } x \in \mathcal{D}}} \sum_{\substack{\text{positions} \\ n}} w_{n',n}(x|y)\, x[n], \qquad w_{n',n}(x|y) \propto \exp\Big(- \frac{\big\|Q_{n'}y - Q_nAx\big\|^2_{Q_n}}{2\sigma^2} \Big)$$

Similar patches $\Rightarrow$ large w. The weights add up to 1. Metric: $\|v\|_{Q_n} \stackrel{\text{def}}{=} \|Q_n^+ v\|$, with $Q_n = \Pi_n B$.

The formula: compare patches, average centres



$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{\substack{\text{training} \\ \text{images } x \in \mathcal{D}}} \sum_{\substack{\text{positions} \\ n}} w_{n',n}(x|y)\; x[n], \qquad w_{n',n}(x|y) \propto \exp\Big(- \frac{\big\| Q_{n'}y - Q_nAx \big\|^2_{Q_n}}{2\sigma^2} \Big)$$

Similar patches $\Rightarrow$ large w. The weights add up to 1. Metric: $\|v\|_{Q_n} \overset{\text{def}}{=} \|Q_n^+ v\|$, with $Q_n = \Pi_n B$.

Denoising specialization: patch similarity becomes explicit

For $A = B = \mathrm{Id}$, the weight used to reconstruct pixel n' becomes

$$w_{n',n}(x \mid y) \propto \exp\left(-\frac{\|y[\omega_{n'}] - x[\omega_n]\|^2}{2\sigma^2}\right).$$

- Compare the observed patch around n' with every training patch.
- Average their center pixels using the Gaussian similarity weights.
- $\sigma \to 0$: nearest-patch lookup; $\sigma \to \infty$: global averaging.

Classical non-local means (Buades et al. 2005)



patch of the pixel to denoise its most similar patches, searched in the **same noisy image**

A data-driven counterpart of non-local means: the look-alikes are searched in the training set, and their clean center pixels are averaged.

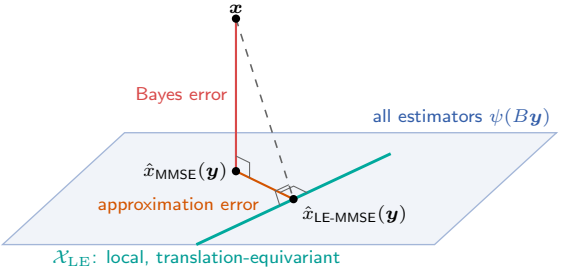
Properties of the LE-MMSE estimator

It is the orthogonal projection of the MMSE onto $\mathcal{X}_{\mathrm{LE}} \overset{\text{def}}{=} \{\phi(B\boldsymbol{y}) : \phi \in \mathcal{M}_{\mathrm{LE}}\}$

$$\mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right] = \underbrace{\mathbb{E}\left[\|\hat{x}_{\mathsf{MMSE}}(\boldsymbol{y}) - \boldsymbol{x}\|^2\right]}_{\text{Bayes error}} + \underbrace{\mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \hat{x}_{\mathsf{MMSE}}(\boldsymbol{y})\|^2\right]}_{\text{approximation error}}$$

$\Rightarrow$ projecting $\boldsymbol{x}$ onto $\mathcal{X}_{\mathrm{LE}}$ **directly**, or **through the MMSE**, gives the same point:

$$\boxed{\hat{x}_{\text{LE-MMSE}}(\boldsymbol{y}) = \mathrm{proj}_{\mathcal{X}_{\mathrm{LE}}}(\boldsymbol{x}) = \mathrm{proj}_{\mathcal{X}_{\mathrm{LE}}}\left(\hat{x}_{\mathsf{MMSE}}(\boldsymbol{y})\right)}$$



. . . but it is *not* a posterior mean

Denoising ($A = B = \mathrm{Id}$), generic dataset $\mathcal{D}$: there is **no prior** $p(x)$ such that $\hat{x}_{\text{LE-MMSE}}(y) = \mathbb{E}[x|\boldsymbol{y} = y]$.

Proof idea. Tweedie: $\mathbb{E}[x|y = y] = y + \sigma^2 \nabla \log p_y(y) \Rightarrow$ the Jacobian of a posterior mean is **symmetric**.
But $\partial_{y_m} \hat{x}_{\text{LE-MMSE}}(y)[n]$ only sees the patch $y[\omega_n]$, and $\partial_{y_n} \hat{x}_{\text{LE-MMSE}}(y)[m]$ only $y[\omega_m]$: **no symmetry** for generic $\mathcal{D}$.

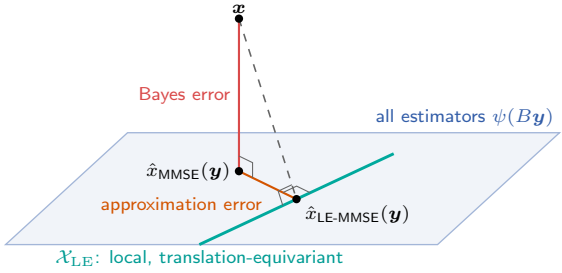
Properties of the LE-MMSE estimator

It is the orthogonal projection of the MMSE onto $\mathcal{X}_{\mathrm{LE}} \overset{\text{def}}{=} \{\phi(B\boldsymbol{y}) : \phi \in \mathcal{M}_{\mathrm{LE}}\}$

$$\mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \boldsymbol{x}\|^2\right] = \underbrace{\mathbb{E}\left[\|\hat{x}_{\mathsf{MMSE}}(\boldsymbol{y}) - \boldsymbol{x}\|^2\right]}_{\text{Bayes error}} + \underbrace{\mathbb{E}\left[\|\phi(B\boldsymbol{y}) - \hat{x}_{\mathsf{MMSE}}(\boldsymbol{y})\|^2\right]}_{\text{approximation error}}$$

$\Rightarrow$ projecting $\boldsymbol{x}$ onto $\mathcal{X}_{\mathrm{LE}}$ **directly**, or **through the MMSE**, gives the same point:

$$\boxed{\hat{x}_{\text{LE-MMSE}}(\boldsymbol{y}) = \mathrm{proj}_{\mathcal{X}_{\mathrm{LE}}}(\boldsymbol{x}) = \mathrm{proj}_{\mathcal{X}_{\mathrm{LE}}}\left(\hat{x}_{\mathsf{MMSE}}(\boldsymbol{y})\right)}$$



. . . but it is *not* a posterior mean

Denoising ($A = B = \mathrm{Id}$), generic dataset $\mathcal{D}$: there is **no prior** $p(\boldsymbol{x})$ such that $\hat{x}_{\text{LE-MMSE}}(y) = \mathbb{E}\left[\boldsymbol{x}|\boldsymbol{y} = y\right]$.

Proof idea. Tweedie: $\mathbb{E}\left[\boldsymbol{x}|\boldsymbol{y} = y\right] = y + \sigma^2 \nabla \log p_{\boldsymbol{y}}(y) \Rightarrow$ the Jacobian of a posterior mean is **symmetric**.
But $\partial_{y_m} \hat{x}_{\text{LE-MMSE}}(y)[n]$ only sees the patch $y[\omega_n]$, and $\partial_{y_n} \hat{x}_{\text{LE-MMSE}}(y)[m]$ only $y[\omega_m]$: **no symmetry** for generic $\mathcal{D}$.

Physics-agnostic versus physics-informed estimator

The estimator has form:

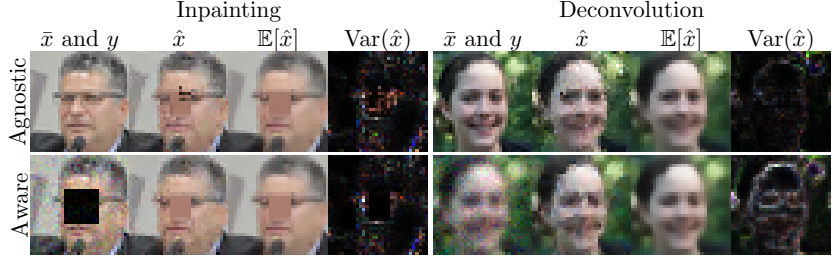
$$\hat{x}_{\text{LE-MMSE}}(y) = \phi^\star(By) \qquad \text{where } y = A\bar{x} + e$$

Let $\Delta_{n',n}(\bar{x}, x) \overset{\text{def}}{=} (BA\bar{x})[\omega_{n'}] - (BAx)[\omega_n]$ and $Q_n = \Pi_n B$, the weights can be rewritten as:

$$w_{n',n}(x|y) \propto \exp\left(-\frac{\eta^2}{2\sigma^2}\right) \quad \text{with} \quad \eta \overset{\text{def}}{=} \left\| Q_n^+ \Delta_{n',n}(\bar{x}, x) + Q_n^+ Q_{n'} e \right\|. \tag{2}$$

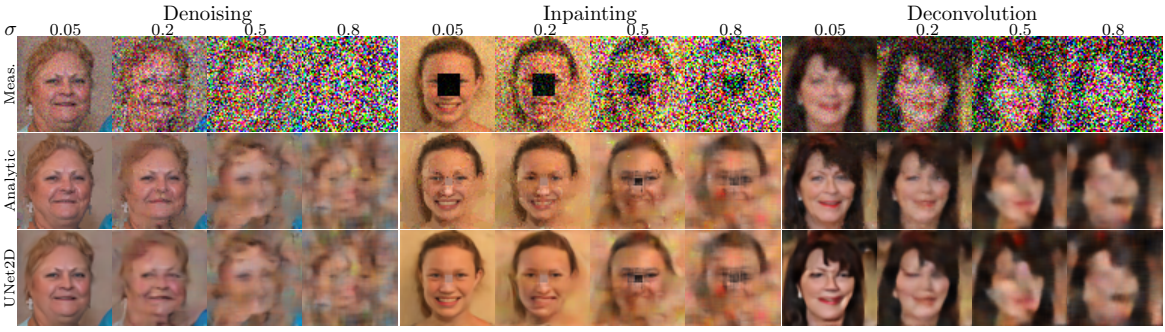
The action of B:

- $Q_n^+ \Delta_{n',n}(\bar{x}, x)$: discriminate similar / dissimilar patches
- $Q_n^+ Q_{n'} e$: amplify / attenuate noise



Qualitative results: FFHQ at 64×64

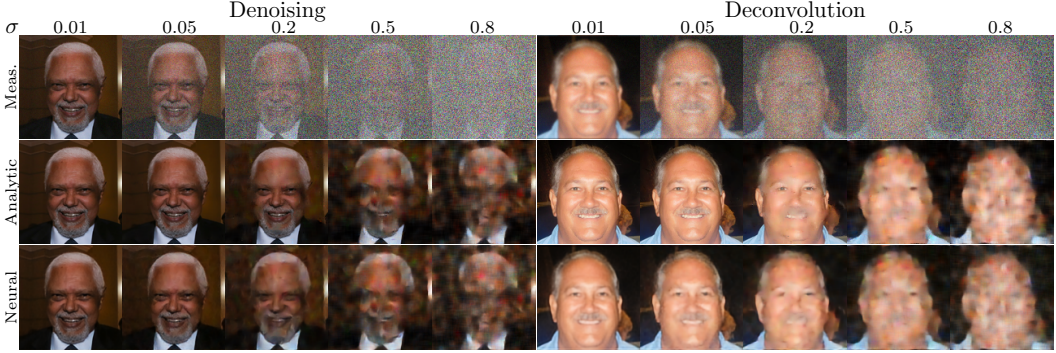
Test images 10,000 training images • UNet2D, 11×11 receptive field • about 40 million training patches



Strong visual agreement across denoising, inpainting, and deconvolution.

Qualitative results: FFHQ at 128×128

Training images 3,000 training images • ResNet, 11×11 receptive field • about 50 million training patches



Strong visual agreement between the network and the formula.

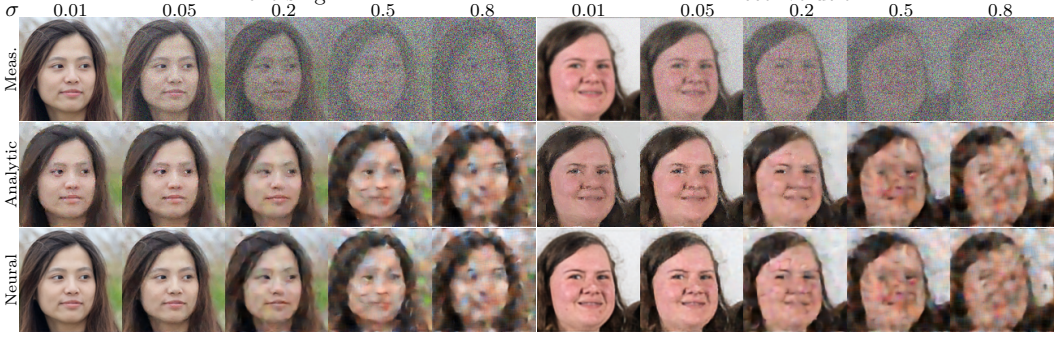
Test alignment PSNR (dB)

	$\sigma = .05$	$\sigma = .5$
Denoising	25.03	29.33
Deblurring	24.72	30.83

Qualitative results: FFHQ at 128×128

Test images

3,000 training images • ResNet, 11×11 receptive field • about 50 million training patches



Strong visual agreement between the network and the formula.

Test alignment PSNR (dB)

	$\sigma = .05$	$\sigma = .5$
Denoising	25.03	29.33
Deblurring	24.72	30.83

The analytical formula matches the output of CNNs sytematically

- Architectures: UNet, ResNet, and PatchMLP, **modified to be local and translation equivariant** (receptive field = patch, circular padding).
- Data: FashionMNIST, CIFAR-10, and FFHQ; tasks: denoising, deblurring, and inpainting.
- Metric: PSNR between the **neural output** and the **analytical output**; this is an alignment metric, not reconstruction PSNR.

The formula closely matches the networks across architectures, datasets, and tasks. (PSNR = 25–40 dB)

Additional inductive bias: spectral bias

- Trained CNNs prefer **low-frequency functions** (spectral bias), and our constraints do not model this.
- A proxy is **randomized smoothing**, which averages the formula over small input perturbations:

$$\hat{x}^{\text{smooth}}_{\text{LE-MMSE}}(y) \overset{\text{def}}{=} \mathbb{E}\left[\hat{x}_{\text{LE-MMSE}}(y + \epsilon z)\right], \qquad z \sim \mathcal{N}(0, \mathrm{Id}).$$

- It acts as a nonlinear low-pass filter, and mirrors training, where each epoch sees a new noise realization.

PSNR gain of the smoothed over the original LE-MMSE ($\epsilon = 0.05$, 50 Monte-Carlo samples) • FFHQ-32, $P = 5 \times 5$, $\sigma = 0.05$

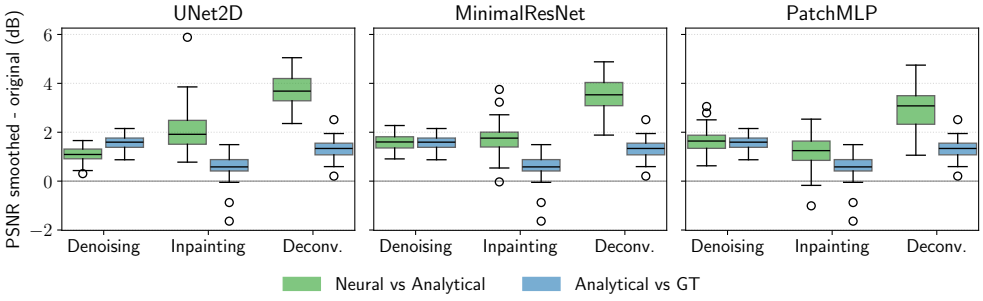
Smoothing brings the formula closer to the networks (+1–3 dB) and to the ground truth (about +1 dB).

Additional inductive bias: spectral bias

- Trained CNNs prefer **low-frequency functions** (spectral bias), and our constraints do not model this.
- A proxy is **randomized smoothing**, which averages the formula over small input perturbations:

$$\hat{x}^{\text{smooth}}_{\text{LE-MMSE}}(y) \overset{\text{def}}{=} \mathbb{E}\left[\hat{x}_{\text{LE-MMSE}}(y + \epsilon z)\right], \qquad z \sim \mathcal{N}(0, \mathrm{Id}).$$

- It acts as a nonlinear low-pass filter, and mirrors training, where each epoch sees a new noise realization.



PSNR gain of the smoothed over the original LE-MMSE ($\epsilon = 0.05$, 50 Monte-Carlo samples) • FFHQ-32, $P = 5 \times 5$, $\sigma = 0.05$

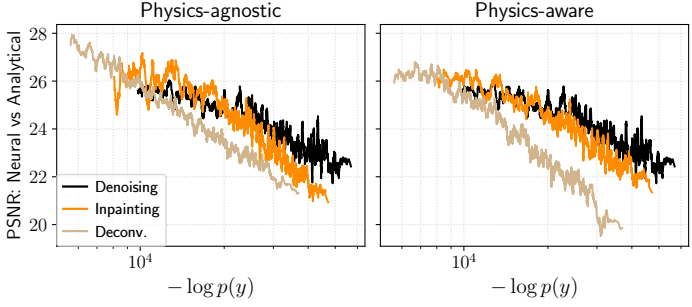
Smoothing brings the formula closer to the networks (+1–3 dB) and to the ground truth (about +1 dB).

Alignment depends on the input distribution

Both the network and the theoretical estimator are optimized under the **measurement distribution**

$$p_{\boldsymbol{y}}(y) = \frac{1}{D} \sum_{x \in \mathcal{D}} \mathcal{N}\left(y; Ax, \sigma^2 \mathrm{Id}\right).$$

- Agreement is strongest where training inputs provide good probability coverage.
- For LE-MMSE, the relevant local input is the transformed patch $\Pi_n B y$; larger patches are harder to cover densely.
- Across the reported noise levels, greater overlap of the Gaussian components can improve coverage.



At fixed $\sigma = 0.05$, lower density is associated with lower neural–analytic PSNR.

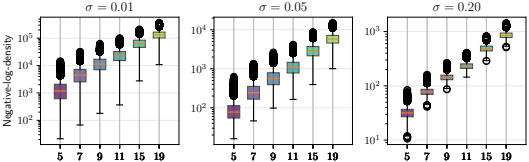
Larger patches: sparser coverage, weaker agreement

Larger patches ⇒ lower patch density

The weights $w_{n',n}$ are normalized by the **patch density** (here for denoising):

$$p(y[\omega]) = \frac{1}{N|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{n=1}^{N} \mathcal{N}\left(y[\omega];\, x[\omega_n],\, \sigma^2 \mathrm{Id}_P\right).$$

It mixes $N|\mathcal{D}| \approx 10^7$ Gaussians in $\mathbb{R}^P$: for a fixed dataset, a larger P leaves them **sparser**.



$-\log p$ of test patches vs. patch size, from 5×5 to 19×19

Lower density ⇒ weaker agreement

PSNR between UNet2D and the formula, for receptive fields $P = 5 \times 5$ to 11×11.
Test set, small σ: PSNR falls as P grows (about 5 dB from 5×5 to 11×11).
Large σ: the Gaussians overlap, the density stays high, and all sizes agree well.

Smaller receptive fields cover the patch space more densely, so the network follows the formula more closely.

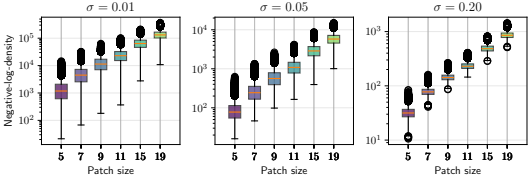
Larger patches: sparser coverage, weaker agreement

Larger patches ⇒ lower patch density

The weights $w_{n',n}$ are normalized by the **patch density** (here for denoising):

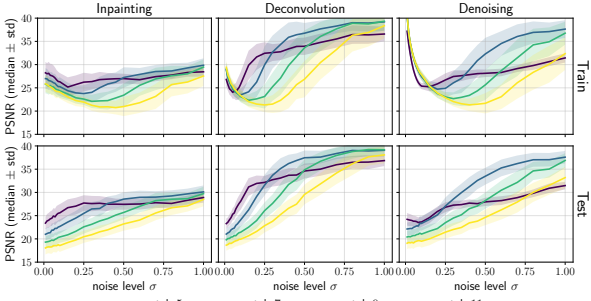
$$p(y[\omega]) = \frac{1}{N|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{n=1}^{N} \mathcal{N}\left(y[\omega];\, x[\omega_n],\, \sigma^2 \mathrm{Id}_P\right).$$

It mixes $N|\mathcal{D}| \approx 10^7$ Gaussians in $\mathbb{R}^P$: for a fixed dataset, a larger P leaves them **sparser**.



$-\log p$ of test patches vs. patch size, from 5×5 to 19×19

Lower density ⇒ weaker agreement



PSNR between UNet2D and the formula, for receptive fields $P = 5 \times 5$ to 11×11.
Test set, small σ: PSNR falls as P grows (about 5 dB from 5×5 to 11×11).
Large σ: the Gaussians overlap, the density stays high, and all sizes agree well.

Smaller receptive fields cover the patch space more densely, so the network follows the formula more closely.

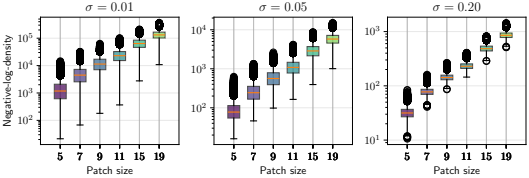
Larger patches: sparser coverage, weaker agreement

Larger patches ⇒ lower patch density

The weights $w_{n',n}$ are normalized by the **patch density** (here for denoising):

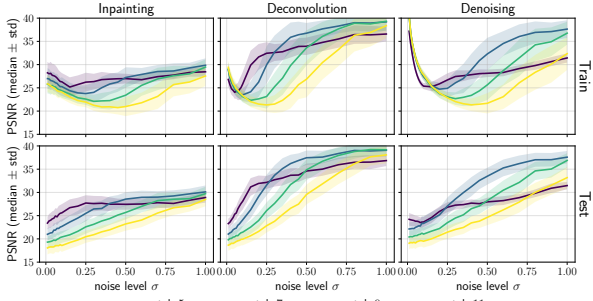
$$p(y[\omega]) = \frac{1}{N|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{n=1}^{N} \mathcal{N}\left(y[\omega];\, x[\omega_n],\, \sigma^2 \mathrm{Id}_P\right).$$

It mixes $N|\mathcal{D}| \approx 10^7$ Gaussians in $\mathbb{R}^P$: for a fixed dataset, a larger P leaves them **sparser**.



$-\log p$ of test patches vs. patch size, from 5×5 to 19×19

Lower density ⇒ weaker agreement



PSNR between UNet2D and the formula, for receptive fields $P = 5 \times 5$ to 11×11.
Test set, small σ: PSNR falls as P grows (about 5 dB from 5×5 to 11×11).
Large σ: the Gaussians overlap, the density stays high, and all sizes agree well.

Smaller receptive fields cover the patch space more densely, so the network follows the formula more closely.

Recap: an analytic theory of CNN solvers

So far

- MSE-trained CNNs that are **local and translation equivariant** behave like the best estimator of this class: the **LE-MMSE**.
- It has a **closed form**: compare patches, average their centres $\Rightarrow$ a **patchwork** of training patches.
- It **matches trained CNNs** across architectures, datasets and inverse problems.
- **Limitations:** in the low-density region, the analytical formula can disagree with the network, and the network generalizes better.

Published at ICML 2026

An analytic theory of convolutional neural network inverse problems solvers

M.-H. Nguyen, Q-B. Do, E. Pauwels, P. Weiss



arXiv

Next (work in progress):

By **analyzing these formulas**, we can gain insights to better understand the **learning problem** when solving inverse problems.

Recap: an analytic theory of CNN solvers

So far

- MSE-trained CNNs that are **local and translation equivariant** behave like the best estimator of this class: the **LE-MMSE**.
- It has a **closed form**: compare patches, average their centres $\Rightarrow$ a **patchwork** of training patches.
- It **matches trained CNNs** across architectures, datasets and inverse problems.
- **Limitations:** in the low-density region, the analytical formula can disagree with the network, and the network generalizes better.

Published at ICML 2026

An analytic theory of convolutional neural network inverse problems solvers

M.-H. Nguyen, Q-B. Do, E. Pauwels, P. Weiss



arXiv

Next (work in progress):

By **analyzing these formulas**, we can gain insights to better understand the **learning problem** when solving inverse problems.

When does training data allow accurate recovery?

Work in progress: recovery guarantees for empirical MMSE

Recall: $y = A(\bar{x}) + e$

$$\|\phi_{\theta^\star}(y) - \bar{x}\| \quad \leq \quad \|\phi_{\theta^\star}(y) - \hat{x}(y)\| \quad + \quad \|\hat{x}(y) - \bar{x}\|$$

neural net vs. ground truth

Shown (empirically)
neural net vs. formula
CNN $\approx$ LE-MMSE

→ Next: this part
formula vs. ground truth
recovery guarantees

The learning problem behind:

- How well does the training set represent an **unseen target**?
- Can the measurements distinguish it from **incorrect explanations**?
- How does adding training images change the recovery error?

Goal: connect recovery error to the geometry of the data distribution and the operator.

The nearest example and measurement ambiguity

Fix a target $\bar{x}$, a finite dataset $\mathcal{D}$, and a linear operator A.

Restricted injectivity

$$\alpha(\bar{x}) = \inf_{x \in \mathcal{D} \setminus \{\bar{x}\}} \frac{\|A(x - \bar{x})\|}{\|x - \bar{x}\|}$$

Smallest singular value along secants $x - \bar{x}$. Small $\alpha \longrightarrow$ distinct signals, similar measurements.

The nearest measured example

$x_\star \in \operatorname*{argmin}_{x \in \mathcal{D}} \|A(x - \bar{x})\|$	the training example whose **measurement** is closest to $A\bar{x}$
$r_\star(\bar{x}) = \|A(x_\star - \bar{x})\|$	**representation error**: distance to the target in measurement space
$\beta(\bar{x}) = \|x_\star - \bar{x}\|$	the actual **signal error** of this example; $\beta(\bar{x}) \leq r_\star(\bar{x})/\alpha(\bar{x})$
$w_\star(y) \propto e^{-\|y - Ax_\star\|^2/(2\sigma^2)}$	its **posterior weight** given y: how much the estimator relies on it

A recovery bound for an arbitrary dataset

Informal theorem: **any** dataset $\mathcal{D}$, **any** target $\bar{x}$ (seen or unseen), **any** perturbation e. Here $y = A\bar{x} + e$ and $\alpha(\bar{x}) > 0$.

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \;\le\; \underbrace{\frac{1}{\alpha(\bar{x})}}_{\text{ambiguity}} \left(\underbrace{r_\star(\bar{x})}_{\text{representation}} + \underbrace{2\,\|e\|}_{\text{noise}} + \underbrace{\sigma\sqrt{2\log D}}_{\text{dataset size}} \right)$$

- **Representation:** how well the training data represents the target, *seen through* A?
- **Noise:** for Gaussian noise, $\|e\| \le \sigma\big(\sqrt{M} + \sqrt{2\log(1/\delta)}\big)$ with probability at least $1 - \delta$.
- **Dataset size:** the price of averaging; the $D - 1$ other examples share the posterior weight.
- **Ambiguity:** converts measurement distances into signal errors, and amplifies every term.

Small-noise limit: nearest-neighbour

Fixed dataset, unique nearest measured example $x_\star$. As $\sigma \to 0$ and $e \to 0$:

$$\hat{x}_{\mathsf{MMSE}}(y) \to x_\star, \qquad \text{error} \to \beta(\bar{x}) \le \frac{r_\star(\bar{x})}{\alpha(\bar{x})}.$$

The size term is needed in the worst case

Denoising, $e = 0$, target **in** the dataset, the $D - 1$ other examples all at one point, at distance $\sigma\sqrt{2\log(D-1)}$:

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| = \frac{\sigma}{2}\sqrt{2\log(D-1)}.$$

Deterministic and distribution-free: no assumption on the dataset.

A recovery bound for an arbitrary dataset

Informal theorem: **any** dataset $\mathcal{D}$, **any** target $\bar{x}$ (seen or unseen), **any** perturbation e. Here $y = A\bar{x} + e$ and $\alpha(\bar{x}) > 0$.

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \;\le\; \underbrace{\frac{1}{\alpha(\bar{x})}}_{\text{ambiguity}} \left(\underbrace{r_\star(\bar{x})}_{\text{representation}} + \underbrace{2\,\|e\|}_{\text{noise}} + \underbrace{\sigma\sqrt{2\log D}}_{\text{dataset size}} \right)$$

- **Representation:** how well the training data represents the target, *seen through* A?
- **Noise:** for Gaussian noise, $\|e\| \le \sigma\big(\sqrt{M} + \sqrt{2\log(1/\delta)}\big)$ with probability at least $1 - \delta$.
- **Dataset size:** the price of averaging; the $D - 1$ other examples share the posterior weight.
- **Ambiguity:** converts measurement distances into signal errors, and amplifies every term.

Small-noise limit: nearest-neighbour

Fixed dataset, unique nearest measured example $x_\star$. As $\sigma \to 0$ and $e \to 0$:

$$\hat{x}_{\mathsf{MMSE}}(y) \to x_\star, \qquad \text{error} \to \beta(\bar{x}) \le \frac{r_\star(\bar{x})}{\alpha(\bar{x})}.$$

The size term is needed in the worst case

Denoising, $e = 0$, target **in** the dataset, the $D - 1$ other examples all at one point, at distance $\sigma\sqrt{2\log(D-1)}$:

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| = \frac{\sigma}{2}\sqrt{2\log(D-1)}.$$

Deterministic and distribution-free: no assumption on the dataset.

Reading the bound: insights and open questions

The bound separates how many examples we have from where they are.

Insights

- **Two conditions for recovery.**
  Representation: an example close to the target in measurement space.
  Separation: the measurements tell the target apart from the other examples.
- **Adding training images:**
  $r_\star$ can only decrease $\Rightarrow$ better coverage;
  α can only decrease $\Rightarrow$ more competitors;
  $\sqrt{\log D}$ grows slowly: $D = 10^3 \to 10^6$ costs a factor 1.4.
- **Which** images are added matters more than **how many**.

Questions it raises

- Is the $\sqrt{\log D}$ penalty real for typical datasets, or only a worst case?
- More of the same data, or more **diverse** data: which one improves $r_\star/\alpha$?
- How do $r_\star$ and α scale with D for natural images?
- Do these insights transfer to the real CNNs?

More data helps when the representation (coverage) gain beats the loss of separation.

Reading the bound: insights and open questions

The bound separates how many examples we have from where they are.

Insights

- **Two conditions for recovery.**
  Representation: an example close to the target in measurement space.
  Separation: the measurements tell the target apart from the other examples.
- **Adding training images:**
  $r_\star$ can only decrease $\Rightarrow$ better coverage;
  α can only decrease $\Rightarrow$ more competitors;
  $\sqrt{\log D}$ grows slowly: $D = 10^3 \to 10^6$ costs a factor 1.4.
- **Which** images are added matters more than **how many**.

Questions it raises

- Is the $\sqrt{\log D}$ penalty real for typical datasets, or only a worst case?
- More of the same data, or more **diverse** data: which one improves $r_\star/\alpha$?
- How do $r_\star$ and α scale with D for natural images?
- Do these insights transfer to the real CNNs?

More data helps when the representation (coverage) gain beats the loss of separation.

Reading the bound: insights and open questions

The bound separates how many examples we have from where they are.

Insights

- **Two conditions for recovery.**
  Representation: an example close to the target in measurement space.
  Separation: the measurements tell the target apart from the other examples.
- **Adding training images:**
  $r_\star$ can only decrease $\Rightarrow$ better coverage;
  α can only decrease $\Rightarrow$ more competitors;
  $\sqrt{\log D}$ grows slowly: $D = 10^3 \to 10^6$ costs a factor 1.4.
- **Which** images are added matters more than **how many**.

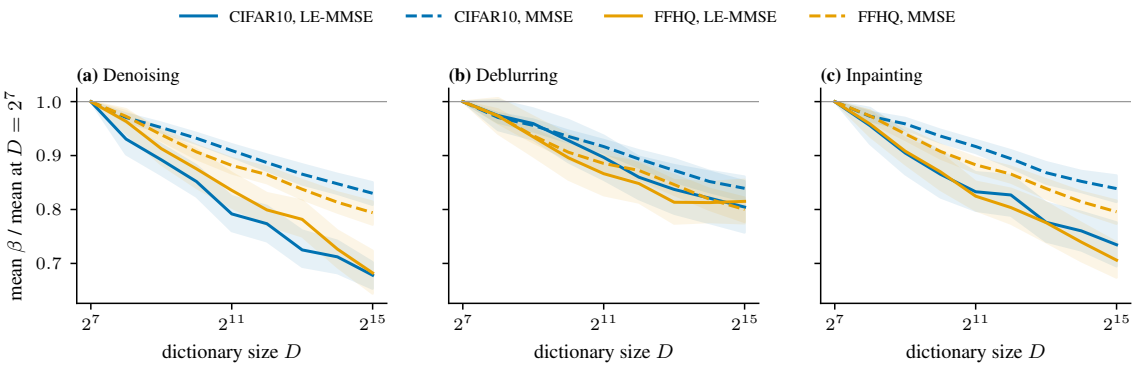
Questions it raises

- Is the $\sqrt{\log D}$ penalty real for typical datasets, or only a worst case?
- More of the same data, or more **diverse** data: which one improves $r_\star/\alpha$?
- How do $r_\star$ and α scale with D for natural images?
- Do these insights transfer to the real CNNs?

More data helps when the representation (coverage) gain beats the loss of separation.

Do more samples improve the nearest example?

Preliminary: CIFAR10 and FFHQ, 32×32 grayscale images; LE-MMSE uses 5×5 patches.



- Mean nearest-example error β, relative to its value at $D = 128$. Below 1 means improvement.
- Mean β decreases across the tested datasets, operators and both estimators.

In these experiments, more images reduce the mean nearest-example error.

Large datasets: the size penalty disappears

At fixed σ, the term $\sigma\sqrt{2\log D}$ grows with D. Does more data hurt?

Same theorem, with a neighbourhood instead of the nearest example: any radius $\rho \geq r_\star(\bar{x})$

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \frac{1}{\alpha(\bar{x})}\left(\rho + 2\,\|e\| + \sigma\sqrt{2\log\frac{D}{D_\rho}}\right), \qquad D_\rho = \#\{x \in \mathcal{D} : \|A(x-\bar{x})\| \leq \rho\}.$$

Consistency (informal): $x_i \overset{\text{iid}}{\sim} \mu$ with compact support $\mathcal{S} \ni \bar{x}$, Gaussian noise, injectivity on the support: $\alpha_{\mathcal{S}}(\bar{x}) > 0$

For every accuracy $\eta > 0$ and $\delta \in (0,1)$, there exist D_0 and $\sigma_0 > 0$ such that

$$D \geq D_0, \quad \sigma \leq \sigma_0 \qquad \Longrightarrow \qquad \|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \eta \quad \text{with probability at least } 1-\delta.$$

Why: the penalty depends on the **fraction** D_ρ/D of examples near the target, not on the dataset size. For large D, this fraction is at least $p/2$ with high probability, where $p > 0$ is the mass of μ near $\bar{x}$.

	Arbitrary dataset (worst case)	**Sampled dataset, large D**
Size dependence	$\sigma\sqrt{2\log D}$	$\sigma\sqrt{2\log(2/p)}$: independent of D
Limit $\sigma \to 0$	error $\to \beta(\bar{x})$, the nearest example	error $\to 0$ as $D \to \infty$

Exact recovery in the limit $D \to \infty$, $\sigma \to 0$.

Large datasets: the size penalty disappears

At fixed σ, the term $\sigma\sqrt{2\log D}$ grows with D. Does more data hurt?

Same theorem, with a neighbourhood instead of the nearest example: any radius $\rho \geq r_\star(\bar{x})$

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \frac{1}{\alpha(\bar{x})}\left(\rho + 2\,\|e\| + \sigma\sqrt{2\log\frac{D}{D_\rho}}\right), \qquad D_\rho = \#\{x \in \mathcal{D} : \|A(x-\bar{x})\| \leq \rho\}.$$

Consistency (informal): $x_i \overset{\text{iid}}{\sim} \mu$ with compact support $\mathcal{S} \ni \bar{x}$, Gaussian noise, injectivity on the support: $\alpha_{\mathcal{S}}(\bar{x}) > 0$

For every accuracy $\eta > 0$ and $\delta \in (0,1)$, there exist D_0 and $\sigma_0 > 0$ such that

$$D \geq D_0, \quad \sigma \leq \sigma_0 \qquad \Longrightarrow \qquad \|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \eta \quad \text{with probability at least } 1-\delta.$$

Why: the penalty depends on the **fraction** D_ρ/D of examples near the target, not on the dataset size. For large D, this fraction is at least $p/2$ with high probability, where $p > 0$ is the mass of μ near $\bar{x}$.

	Arbitrary dataset (worst case)	**Sampled dataset, large D**
Size dependence	$\sigma\sqrt{2\log D}$	$\sigma\sqrt{2\log(2/p)}$: independent of D
Limit $\sigma \to 0$	error $\to \beta(\bar{x})$, the nearest example	error $\to 0$ as $D \to \infty$

Exact recovery in the limit $D \to \infty$, $\sigma \to 0$.

Large datasets: the size penalty disappears

At fixed σ, the term $\sigma\sqrt{2\log D}$ grows with D. Does more data hurt?

Same theorem, with a neighbourhood instead of the nearest example: any radius $\rho \geq r_\star(\bar{x})$

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \frac{1}{\alpha(\bar{x})}\left(\rho + 2\,\|e\| + \sigma\sqrt{2\log\frac{D}{D_\rho}}\right), \qquad D_\rho = \#\{x \in \mathcal{D} : \|A(x-\bar{x})\| \leq \rho\}.$$

Consistency (informal): $x_i \overset{\text{iid}}{\sim} \mu$ with compact support $\mathcal{S} \ni \bar{x}$, Gaussian noise, injectivity on the support: $\alpha_{\mathcal{S}}(\bar{x}) > 0$

For every accuracy $\eta > 0$ and $\delta \in (0,1)$, there exist D_0 and $\sigma_0 > 0$ such that

$$D \geq D_0, \quad \sigma \leq \sigma_0 \qquad \Longrightarrow \qquad \|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \eta \quad \text{with probability at least } 1-\delta.$$

Why: the penalty depends on the **fraction** D_ρ/D of examples near the target, not on the dataset size. For large D, this fraction is at least $p/2$ with high probability, where $p > 0$ is the mass of μ near $\bar{x}$.

	Arbitrary dataset (worst case)	**Sampled dataset, large D**
Size dependence	$\sigma\sqrt{2\log D}$	$\sigma\sqrt{2\log(2/p)}$: independent of D
Limit $\sigma \to 0$	error $\to \beta(\bar{x})$, the nearest example	error $\to 0$ as $D \to \infty$

Exact recovery in the limit $D \to \infty$, $\sigma \to 0$.

A tractable model: subspace data, random measurements

The bounds involve abstract quantities: α, $r_\star$, the mass near the target. Can we compute them?

Low-dimensional signals

$$x_i = U z_i, \qquad z_i \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{B}_d), \qquad \bar{x} = U\bar{z}.$$

$U \in \mathbb{R}^{N\times d}$ has orthonormal columns. $\mathbb{B}_d$ is the unit ball; $\bar{z} \in \mathbb{B}_d$ is fixed.

Random measurements

$$y = A\bar{x} + e, \qquad A_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/M).$$

Draw A independently of the data, then keep it fixed and known.

Why this model?

- **Low intrinsic dimension** $d \ll N$: the simplest version of the manifold hypothesis.
- **Random measurements**, as in compressed sensing: A nearly preserves distances on the subspace once $M \gtrsim d$.
- **Everything is explicit:** injectivity, coverage and the mass near the target depend only on D, d and M.

A simple model: explicit enough to compute, rich enough to study how data and measurements interact in the MMSE.

A tractable model: subspace data, random measurements

The bounds involve abstract quantities: α, $r_\star$, the mass near the target. Can we compute them?

Low-dimensional signals

$$x_i = Uz_i, \qquad z_i \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{B}_d), \qquad \bar{x} = U\bar{z}.$$

$U \in \mathbb{R}^{N\times d}$ has orthonormal columns. $\mathbb{B}_d$ is the unit ball; $\bar{z} \in \mathbb{B}_d$ is fixed.

Random measurements

$$y = A\bar{x} + e, \qquad A_{ij} \overset{\text{iid}}{\sim} \mathcal{N}\left(0, 1/M\right).$$

Draw A independently of the data, then keep it fixed and known.

Why this model?

- **Low intrinsic dimension** $d \ll N$: the simplest version of the manifold hypothesis.
- **Random measurements**, as in compressed sensing: A nearly preserves distances on the subspace once $M \gtrsim d$.
- **Everything is explicit:** injectivity, coverage and the mass near the target depend only on D, d and M.

A simple model: explicit enough to compute, rich enough to study how data and measurements interact in the MMSE.

Subspace geometry: injectivity and coverage

For sufficiently many measurements ($a_d > 0$), with high probability over A and the samples:

δ_D: sampling failure probability.

Injectivity stays bounded below

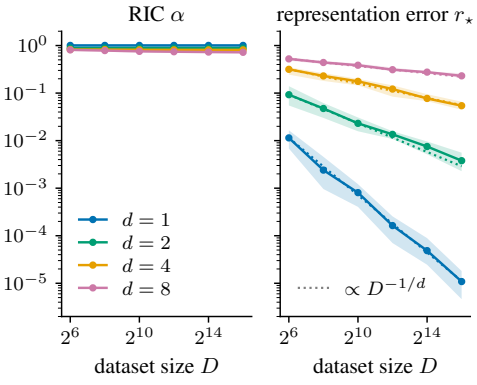
$$\alpha(\bar{x}) \geq a_d > 0.$$

A floor **independent of dataset size**.

Coverage improves at scale $D^{-1/d}$

$$r_\star(\bar{x}) \leq 2b_d \left(\frac{\log(1/\delta_D)}{D} \right)^{1/d}.$$

Higher intrinsic dimension $\longrightarrow$ slower improvement.



From MMSE to LE-MMSE: towards a recovery theory for CNNs

Same structure $\Rightarrow$ the results transfer directly to the LE-MMSE

MMSE: weighted average of training *images*

$$\hat{x}_{\text{MMSE}}(y) = \sum_{x \in \mathcal{D}} w(x|y)\, x$$

$$w(x|y) \propto \exp\Big(- \frac{\|y - Ax\|^2}{2\sigma^2} \Big)$$

LE-MMSE: weighted average of training *patches*

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{x \in \mathcal{D}} \sum_{n} w_{n',n}(x|y)\, x[n]$$

$$w_{n',n}(x|y) \propto \exp\Big(- \frac{\|y[\omega_{n'}] - Ax[\omega_n]\|^2}{2\sigma^2} \Big)$$

The same bounds hold **pixel-wise**, with **patches** in place of images.

. . . which lets us study the stability of CNNs, through their LE-MMSE approximation

$$\|\phi_{\theta^\star}(y) - \bar{x}\| \;\leq\; \underbrace{\|\phi_{\theta^\star}(y) - \hat{x}_{\text{LE-MMSE}}(y)\|}_{\text{small: CNN} \approx \text{LE-MMSE (empirically)}} \;+\; \underbrace{\|\hat{x}_{\text{LE-MMSE}}(y) - \bar{x}\|}_{\text{bounded: this analysis}}$$

From MMSE to LE-MMSE: towards a recovery theory for CNNs

Same structure $\Rightarrow$ the results transfer directly to the LE-MMSE

MMSE: weighted average of training *images*

$$\hat{x}_{\text{MMSE}}(y) = \sum_{x \in \mathcal{D}} w(x|y)\, x$$

$$w(x|y) \propto \exp\Big(- \frac{\|y - Ax\|^2}{2\sigma^2} \Big)$$

LE-MMSE: weighted average of training *patches*

$$\hat{x}_{\text{LE-MMSE}}(y)[n'] = \sum_{x \in \mathcal{D}} \sum_{n} w_{n',n}(x|y)\, x[n]$$

$$w_{n',n}(x|y) \propto \exp\Big(- \frac{\|y[\omega_{n'}] - Ax[\omega_n]\|^2}{2\sigma^2} \Big)$$

The same bounds hold **pixel-wise**, with **patches** in place of images.

. . . which lets us study the stability of CNNs, through their LE-MMSE approximation

$$\|\phi_{\theta^\star}(y) - \bar{x}\| \;\leq\; \underbrace{\|\phi_{\theta^\star}(y) - \hat{x}_{\text{LE-MMSE}}(y)\|}_{\text{small: CNN} \approx \text{LE-MMSE (empirically)}} \;+\; \underbrace{\|\hat{x}_{\text{LE-MMSE}}(y) - \bar{x}\|}_{\text{bounded: this analysis}}$$

Discussion & Perspectives

The bounds turn the effect of training data into geometric questions.

An approach to analyzing MMSE

- A recovery bound identifies **representation, measurement ambiguity and dataset size** as the mechanisms.
- For sampled datasets: recovery is consistent as $D \to \infty$ and $\sigma \to 0$.
- The subspace model makes **coverage, injectivity and the required dataset size** explicit.
- The same bounds hold pixel-wise for **LE-MMSE**, our surrogate for trained CNNs.

Next steps

- **Diverse datasets**: when does adding another population help?
- Use the LE-MMSE bounds to study neural solvers: patch size, operator, back-projection.

Thanks for your attention!