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

Workshop MATHS-IA, 4-6 November 2026, Bordeaux







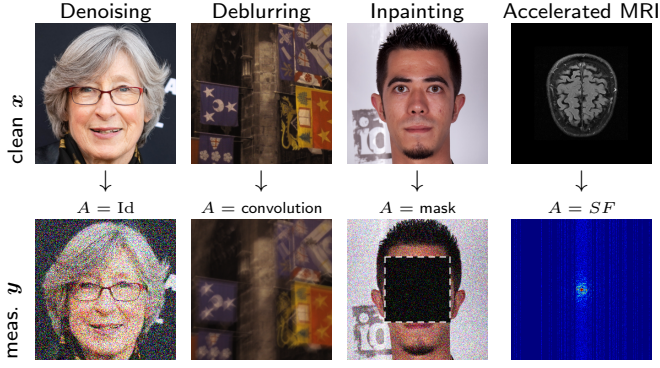
Inverse problems

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**:

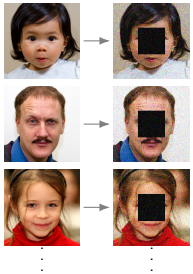
- plug-and-play priors, unrolled networks, generative (diffusion) priors, ...
- **supervised learning**: a network trained on pairs of clean images and measurements.

Today: the supervised learning framework.

Supervised reconstruction

Clean-image dataset and measurements

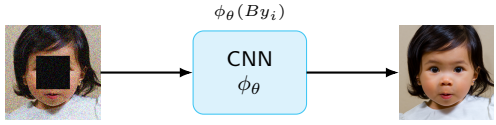
clean x_i measurement y_i



$\mathcal{D} = \{x_1, \ldots, x_D\}$: 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

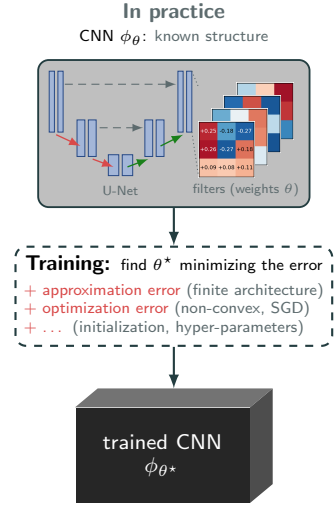
B brings the measurement back to image space, before the network sees it

Training objective (empirical risk minimization)

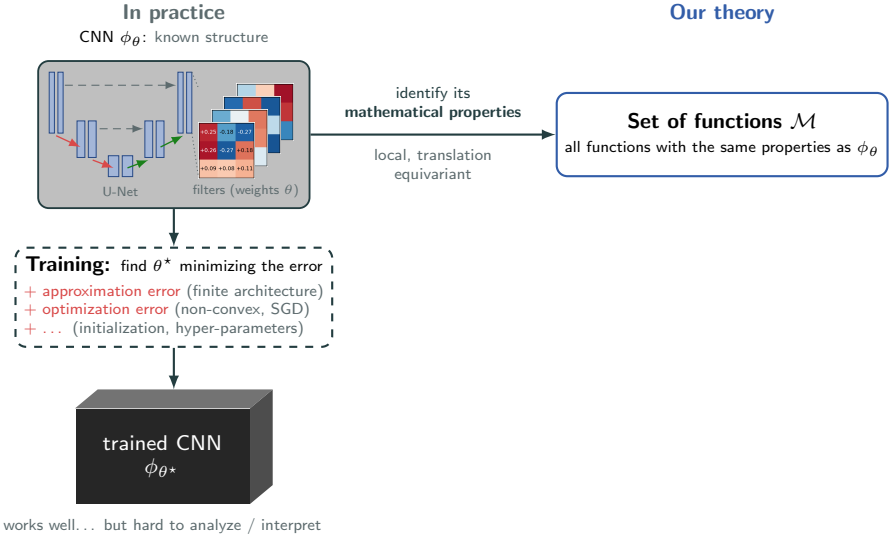
$$\theta^\star \in \operatorname*{argmin}_{\theta \in \Theta} \frac{1}{D} \sum_{i=1}^{D} \|\phi_\theta(By_i) - x_i\|^2, \qquad \text{then} \quad \hat{x}(y) = \phi_{\theta^\star}(By).$$

What does this trained network converge to?

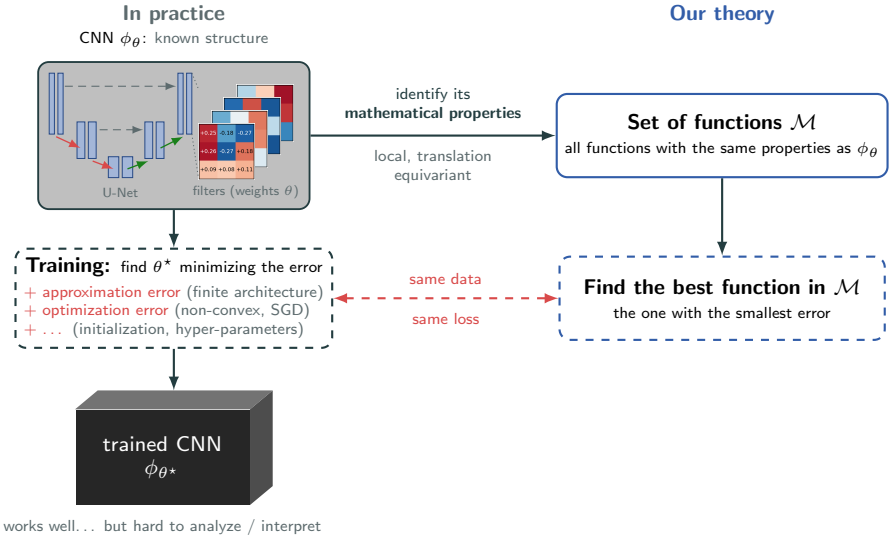
A surrogate analytical model to understand neural networks



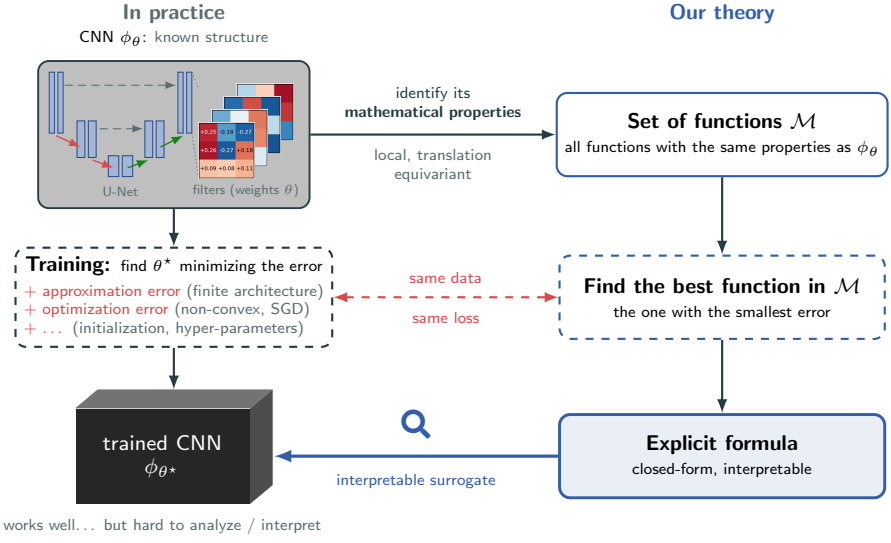
A surrogate analytical model to understand neural networks



A surrogate analytical model to understand neural networks

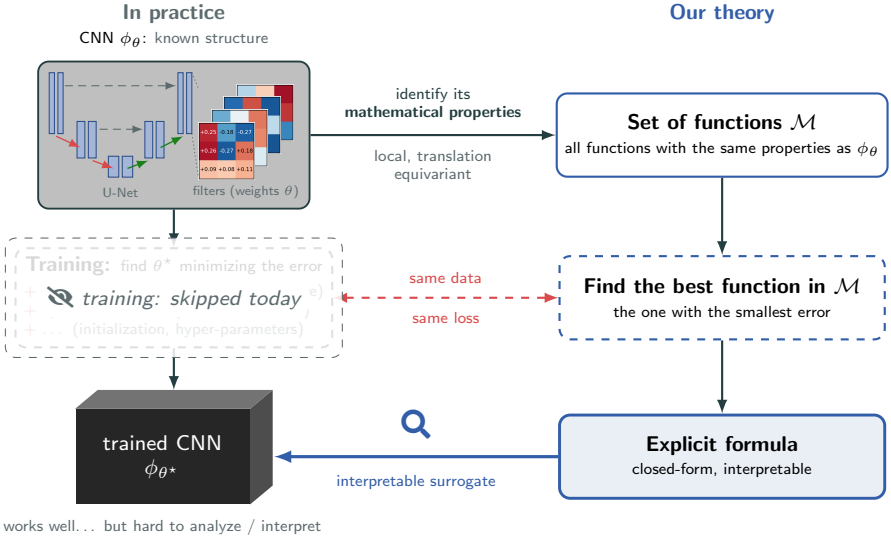


A surrogate analytical model to understand neural networks



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**.

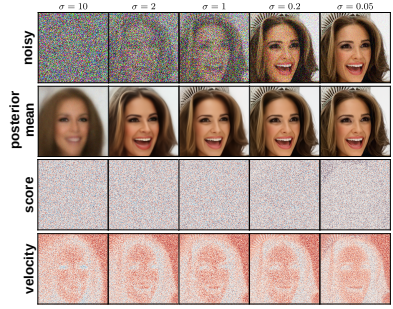
If the neural network is **sufficiently expressive** and **well-trained**, $\phi_{\theta^\star}$ *approximates* the empirical MMSE estimator:

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

Link to generative models

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

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



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 $y \in \mathbb{R}^M$,

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

- 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.

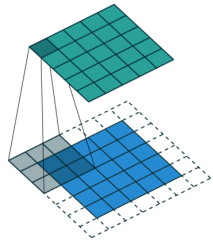
Define:

- $\mathcal{M}_{\mathsf{LE}}$: 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

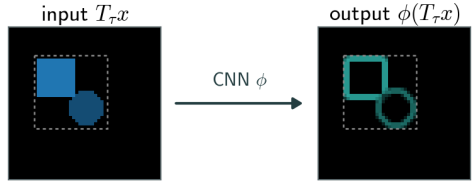
then,

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

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 $\mathrm{conv}(\mathcal{D})$.
- It is the **orthogonal projection** in L^2 of the empirical MMSE estimator onto $\mathcal{X}_{\text{LE}} = \{\phi(B\boldsymbol{y}) : \phi \in \mathcal{M}_{\text{LE}}\}$.

⚠ **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

① **Look around pixel n'**

measurement By



patch $u_{n'} = \Pi_{n'} By$

FFHQ images, denoising ($A = B = \mathrm{Id}$, $\sigma = 0.06$). Illustrative weights; reconstruction: LE-MMSE output with 10 000 training images.

The formula: compare patches, average centres

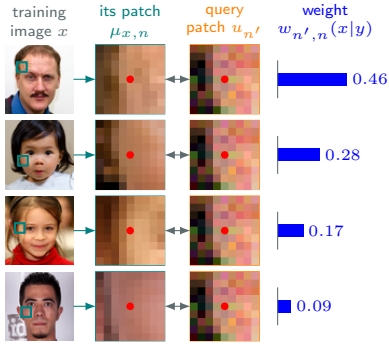
① Look around pixel n'

measurement By



patch $u_{n'} = \Pi_{n'} By$

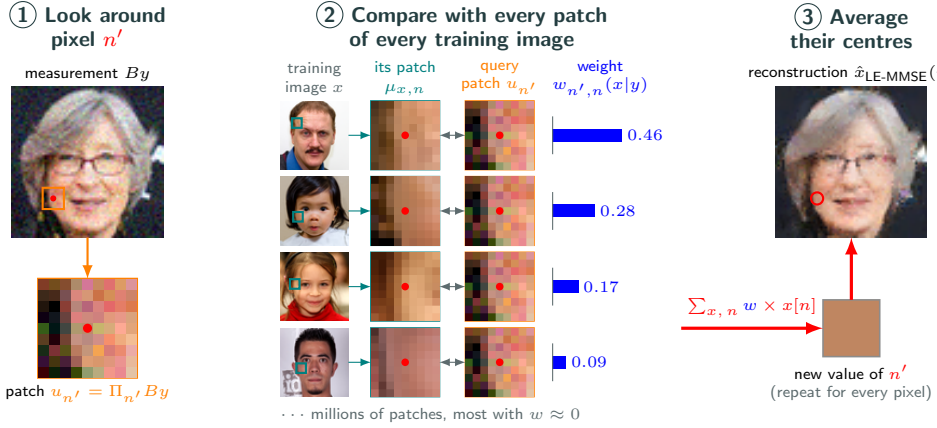
② Compare with every patch of every training image



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

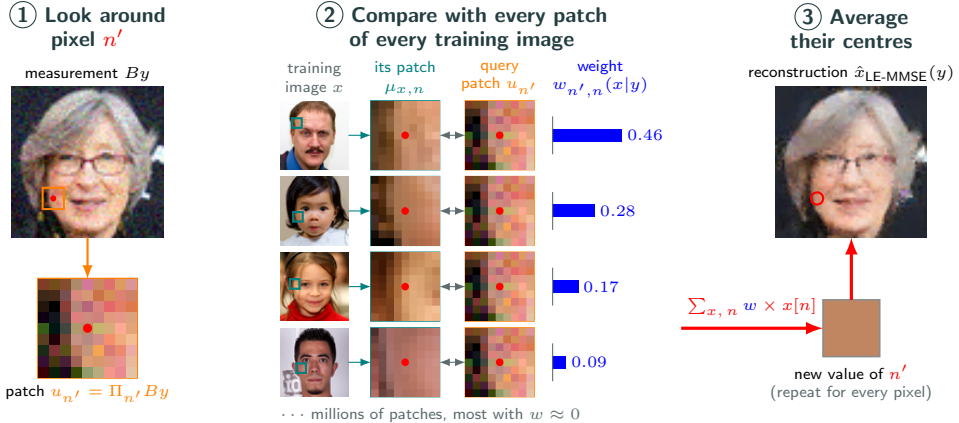
FFHQ images, denoising ($A = B = \mathrm{Id}$, $\sigma = 0.06$). Illustrative weights; reconstruction: LE-MMSE output with 10 000 training images.

The formula: compare patches, average centres



FFHQ images, denoising ($A = B = \mathrm{Id}$, $\sigma = 0.06$). Illustrative weights; reconstruction: LE-MMSE output with 10 000 training images.

The formula: compare patches, average centres



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

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

FFHQ images, denoising ($A = B = \mathrm{Id}$, $\sigma = 0.06$). Illustrative weights; reconstruction: LE-MMSE output with 10 000 training images.

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)

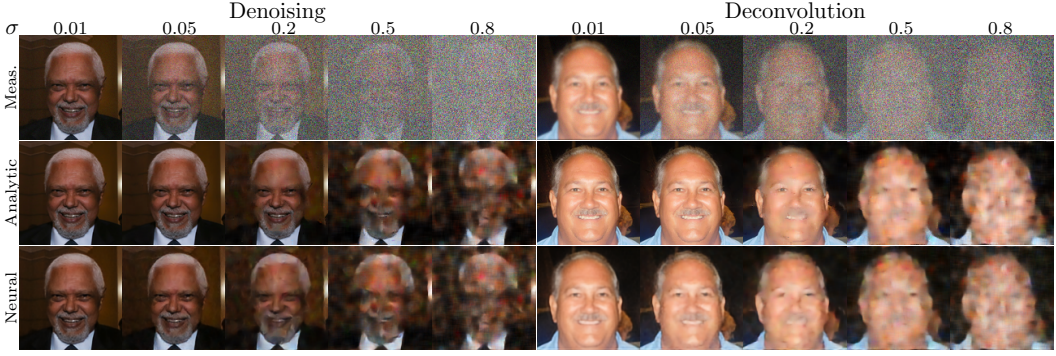


■ patch of the pixel to denoise ■ its most similar patches, searched in the **same noisy image**

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

Qualitative results: FFHQ at 128×128

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

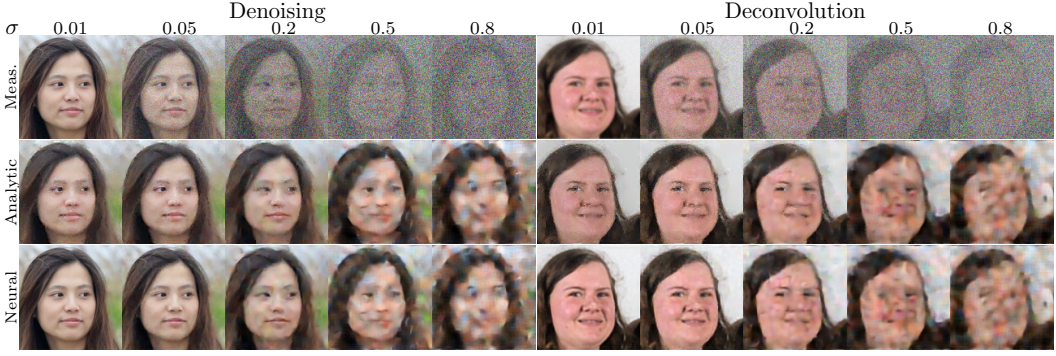


Strong visual agreement between the network and the formula.

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.

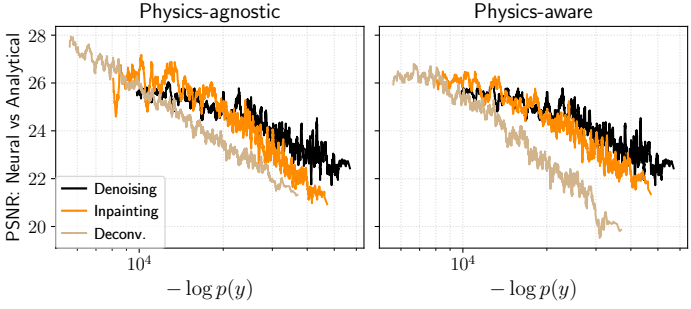
The formula closely matches the networks across architectures, datasets, and tasks. (PSNR = 25–40 dB)

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 $\Pi_n By$; 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.

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

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

Analyzing MMSE through a recovery bound

Worst-case bound: $D \geq 3$, $\sigma > 0$, $0 < \alpha(\bar{x}) < \infty$, and any $y = A\bar{x} + e$

$$\|\hat{x}_{\mathsf{MMSE}}(y) - \bar{x}\| \leq \underbrace{\beta(\bar{x})}_{\text{nearest example}} + \underbrace{(1 - w_\star(y)) \left(\frac{r_\star(\bar{x})}{\alpha(\bar{x})} - \beta(\bar{x}) \right)}_{\text{remaining geometry}} + \underbrace{\frac{2\,\|e\|}{\alpha(\bar{x})}}_{\text{noise}} + \underbrace{\frac{\sigma\sqrt{2\log(D-1)}}{\alpha(\bar{x})}}_{\text{dataset bias}}.$$

- **Nearest example:** how well the signal is represented in the dataset ?
- **Remaining geometry:** how much posterior weight lies away from $x_\star$?
- **Noise:** how strongly does measurement ambiguity amplify perturbations?
- **Dataset bias:** an explicit $\sqrt{\log(D-1)}$ dependence at fixed α, σ.

Small-noise limit: the nearest example determines the error

Fixed dataset, unique nearest measured example $x_\star$, and $e = \sigma z$ with fixed z:

$$\sigma \to 0: \qquad \hat{x}_{\mathsf{MMSE}}(y) \to x_\star, \qquad \text{error and bound} \to \beta(\bar{x}).$$

The bound identifies mechanisms to analyze, even when it is loose.

A similar pixel-wise bound holds for **LE-MMSE**, with local geometry and noise.

What changes when we add training images?

The bound separates sample count from sample geometry.

Sample count: a slow explicit dependence

$$\frac{\sigma\sqrt{2\log(D-1)}}{\alpha(\bar{x})}$$

At fixed α, σ, increasing D from 10^3 to 10^6 multiplies this term by only ≈ 1.4.

Sample geometry: what should we study?

- **Coverage:** do added samples reduce the nearest-example error β?
- **Ambiguity:** how much do they reduce α?

Adding samples cannot increase $r_\star$ or α; the signal error β need not decrease.

Why start with the nearest example?

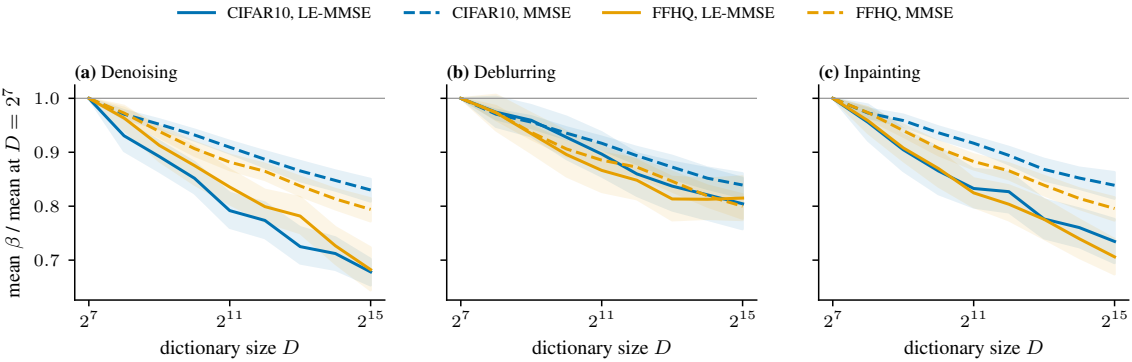
In the small-noise limit, the selected nearest example determines the error.

Its accuracy depends on **which samples are added**.

Study how coverage and measurement ambiguity change together.

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.

A tractable model: subspace data, random measurements

Can we explain these geometric quantities in a tractable model?

Low-dimensional signals

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

$U \in \mathbb{R}^{N \times d}$ 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.

Study coverage and injectivity as functions of sample count and dimension.

Subspace geometry: injectivity and coverage

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

$a_d \|h\| \le \|Ah\| \le b_d \|h\|$ $\quad (h \in \mathrm{range}(U))$: lower and upper measurement gains.
δ_D: sampling failure probability.

Injectivity stays bounded below

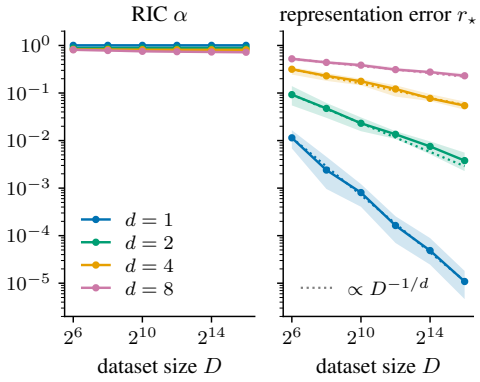
$$\alpha(\bar{x}) \ge a_d > 0.$$

A floor **independent of dataset size**.

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

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

Higher intrinsic dimension $\longrightarrow$ slower improvement.



Subspace geometry: injectivity and coverage

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

$a_d \|h\| \le \|Ah\| \le b_d \|h\|$ $\quad (h \in \mathrm{range}(U))$: lower and upper measurement gains.
δ_D: sampling failure probability.

Injectivity stays bounded below

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

A floor **independent of dataset size**.

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

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

Higher intrinsic dimension $\longrightarrow$ slower improvement.

What the model makes explicit

- Measurement geometry gives an injectivity floor independent of D.
- Intrinsic dimension controls the $D^{-1/d}$ coverage scale.

Next: how does nearby sample mass shape posterior averaging?

Discussion & Perspectives

The bound turns the effect of training data into geometric questions.

An approach to analyzing MMSE

- The bound identifies **coverage, measurement ambiguity and posterior concentration** as mechanisms to study.
- Image experiments examine the **nearest-example term** as the dataset grows.
- The subspace model makes **coverage and injectivity** analytically tractable.

Next steps

- Study how **nearby sample mass** and ambiguity shape posterior averaging at finite noise.
- Use the LE-MMSE surrogate to investigate these mechanisms in neural inverse-problem solvers.

Thanks for your attention!

Subspace geometry: constants and confidence

Measurement gains

For a matrix failure probability $\delta_A \in (0,1)$, set

$$\theta_d = \sqrt{\frac{d}{M}} + \sqrt{\frac{2\log(2/\delta_A)}{M}}, \qquad a_d = 1 - \theta_d, \qquad b_d = 1 + \theta_d.$$

Assume $a_d > 0$. With probability at least $1 - \delta_A$ over A,

$$a_d \|h\| \le \|Ah\| \le b_d \|h\| \qquad \text{for every } h \in \mathrm{range}(U).$$

Thus a_d and b_d bound the smallest and largest singular values of AU.

Sample coverage

For a sampling failure probability $\delta_D \in (0,1)$, jointly with probability at least $1 - \delta_A - \delta_D$,

$$\alpha(\bar{x}) \ge a_d, \qquad r_\star(\bar{x}) \le 2b_d \left(\frac{\log(1/\delta_D)}{D}\right)^{1/d}, \qquad \beta(\bar{x}) \le \frac{r_\star(\bar{x})}{a_d}.$$

At fixed confidence, the gains depend on M, d; the coverage scale is $D^{-1/d}$.