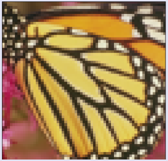
Filling the gaps: a story of priors and conditional probabilities

Thomas Moreau INRIA Saclay

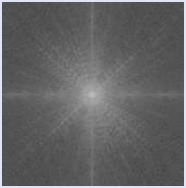
Inverse Problems

Imaging

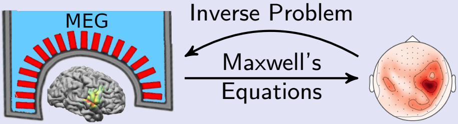



Super-Resolution, Inpainting, Deblurring, ...

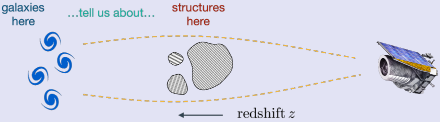
Neuroimaging – MRI



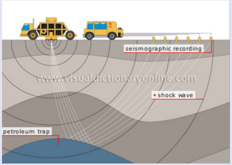

Neuroimaging – M/EEG



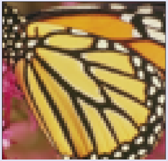
Astrophysics



Seismology – Prospection

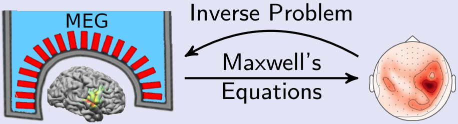


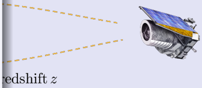
Imaging



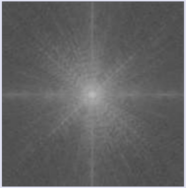
Super-Resolut
Deblu

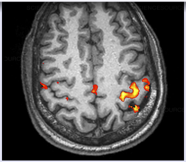
Neuroimaging – M/EEG



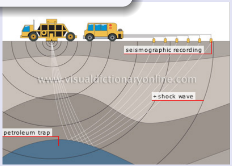


Neuroimag





pection



Forward model:

$$y = Ax + \varepsilon$$

Inverse problem:

find x from y

MAP estimate as a regularized regression problem

$$x^*(y;\theta) = \underset{x}{\operatorname{argmin}} - \log\ p(x|y;\theta) = \underbrace{\frac{1}{2}\|y - Ax\|_2^2}_{-log\ \ p(y|x)} + \underbrace{\mathcal{R}(x;\theta)}_{-log\ \ p(x;\theta)}$$

where $\mathcal{R}$ encodes prior information to select a good/plausible solution

Inverse Problem Resolution

MAP estimate as a regularized regression problem

$$x^*(y;\theta) = \underset{x}{\operatorname{argmin}} -\log\ p(x|y;\theta) = \underbrace{\frac{1}{2}\|y - Ax\|_2^2}_{-\log\ p(y|x)} + \underbrace{\mathcal{R}(x;\theta)}_{-\log\ p(x;\theta)}$$

where $\mathcal{R}$ encodes prior information to select a good/plausible solution

<u>**Common framework:**</u>

- **Efficient solvers:** Forward backward, ADMM, HQS ...

  $\Rightarrow$ But might require many iterations to get good results

- **Flexible:** Can choose many priors – *handpicked, learned, implicit, ...*

  $\Rightarrow$ Quality of the solution depends on the prior's choice $p(x;\theta)$

Defining a the prior $\mathrm{p}(\boldsymbol{x}; \theta)$

Explicitely: choose a log-prior $\mathcal{R}$ promoting certain properties of the signal.

TV, sparsity, wavelets, dictionary, ...

Interpretable, no learning, convergence guarantees

Defining a the prior $\mathrm{p}(\boldsymbol{x}; \theta)$

Explicitely: choose a log-prior $\mathcal{R}$ promoting certain properties of the signal.

TV, sparsity, wavelets, dictionary, ...

Interpretable, no learning, convergence guarantees

Implicitely: we only need the prior's score $\nabla\mathcal{R}$ or its prox

Pnp/RED frameworks with denoisers

Linked to the score with the Tweedie's formula

Defining a the prior $\mathrm{p}(\boldsymbol{x}; \theta)$

Explicitely: choose a log-prior $\mathcal{R}$ promoting certain properties of the signal.

TV, sparsity, wavelets, dictionary, . . .

Interpretable, no learning, convergence guarantees

Implicitely: we only need the prior's score $\nabla\mathcal{R}$ or its prox

Pnp/RED frameworks with denoisers

Linked to the score with the Tweedie's formula

x

$y = x + e$

$\mathrm{prox}_{\mathrm{TV}}(y)$

$\mathrm{DRUNet}(y)$

Prior selection and learning

Supervised learning of the prior

Evaluate the quality of the prior on a dataset $(\mathbf{y}, \mathbf{x})$

$$\min_{\theta} \mathbb{E}_{(\mathbf{y},\mathbf{x})} \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*(\mathbf{y};\theta)\|_2^2 \quad s.t. \quad \mathbf{x}^*(\mathbf{y};\theta) = \operatorname*{argmin}_{\mathbf{y}} -log \; p(\mathbf{x}|\mathbf{y};\theta)$$

Task Agnostic learning

Characterize the distribution of the data $p(\mathbf{x})$ by training a denoiser

$$\min_{\theta} \mathbb{E}_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*(\mathbf{x} + \epsilon;\theta)\|_2^2$$

Self-supervised learning

[Tachella et al., 2022]

Learn with incomplete measurements $\mathbf{y}$ with equivariance, consistency, ...

$$\min_{\theta} \mathbb{E}_{\mathbf{y}=(\mathbf{y}|_1,\mathbf{y}|_2)} \frac{1}{2} \|\mathbf{y}|_2 - \mathbf{A}|_2 \mathbf{x}^*(\mathbf{y}|_1;\theta)\|_2^2$$

Desirable properties of the prior

How to choose the structure of the prior $p(\boldsymbol{x}; \theta)$?

- **Adapted to the task:** allow to recover the information lost by $\boldsymbol{A}$
- **Adapted to the data:** capture the distribution of the data
- **Easy to learn:** few data, fast training, no overfitting
- **Fast to compute:** used in an iterative algorithm
- **Interpretable:** understand what is done

Desirable properties of the prior

How to choose the structure of the prior $p(\boldsymbol{x}; \theta)$?

- **Adapted to the task:** allow to recover the information lost by $\boldsymbol{A}$
- **Adapted to the data:** capture the distribution of the data
- **Easy to learn:** few data, fast training, no overfitting
- **Fast to compute:** used in an iterative algorithm
- **Interpretable:** understand what is done

The role of the structure in unsupervised prior learning

References

- Malézieux, B., Michel, F., **TM**, and Kowalski, M. (2024). Where prior learning can and can't work in unsupervised inverse problems. Preprint

The goal of prior learning

The goal of prior learning is to **recover the information lost by the operator $\boldsymbol{A}$**, while the data fidelity term ensures consistency with the observed measurements.

- **Data fidelity:** Ensures that the solution matches the known measurements (what is observed).
- **Prior:** Completes the missing information (what is lost by $\boldsymbol{A}$), guiding the solution towards plausible images.

$\Rightarrow$ A well-chosen prior needs to recover $P_{\ker \boldsymbol{A}}(\boldsymbol{x})$ from the observations (either $\boldsymbol{y}$ or $\boldsymbol{A}^{\dagger}\boldsymbol{x}$).

Dictionary-based priors: a tool to study prior learning

We consider the problem of learning a dictionary $\boldsymbol{D}$ to solve inverse problems with a sparse prior with a synthesis formulation

$$\boldsymbol{x}^*(\boldsymbol{y}; \boldsymbol{D}) = \boldsymbol{D} \operatorname*{argmin}_{\boldsymbol{z}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{D}\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

- Explicit prior parameterization with the dictionary $\boldsymbol{D}$
- Can generate data according to the model
- Can study the dynamic of learning $\boldsymbol{D}$

Single measurement dictionary learning

With no extra constraint, if the dictionary is learned in an unsupervised manner, the dictionary cannot recover any information lost by $\boldsymbol{A}$

$$\boldsymbol{D} = \underset{\|\boldsymbol{D}\|_2 \leq 1}{\operatorname{argmin}} \underset{\boldsymbol{z}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{D}\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 \implies \boldsymbol{D} \in \ker(\boldsymbol{A})^\perp$$

$\Rightarrow$ The dictionary is null in the null space of $\boldsymbol{A}$

Therefore, the prior cannot help in solving the inverse problem.

Multiple measurement dictionary learning

When learning with multiplte operators $\boldsymbol{A}_i$, the dictionary can recover some information lost by each operator

$$\boldsymbol{D} = \underset{\|\boldsymbol{D}\|_2 \le 1}{\operatorname{argmin}} \sum_i \underset{\boldsymbol{z}_i}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{A}_i \boldsymbol{D} \boldsymbol{z}_i\|_2^2 + \lambda \|\boldsymbol{z}_i\|_1$$

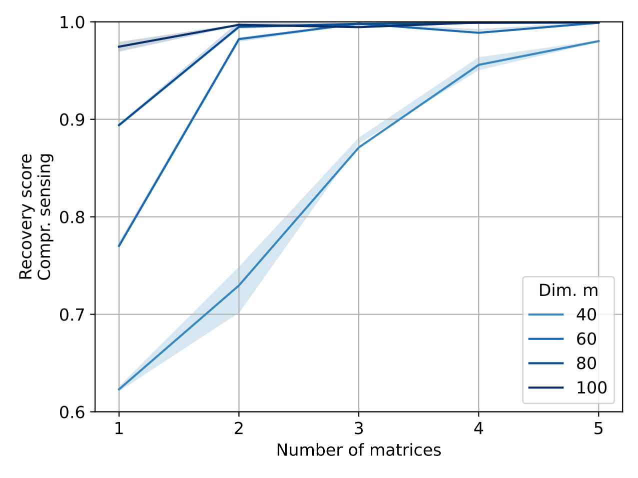
Here, an interesting cases is when the operators $\boldsymbol{A}_i$ are *incomplete* but their union is complete, i.e. $\bigcap_i \ker(\boldsymbol{A}_i) = \{0\}$

In this case, the dictionary can recover information lost by each operator and perform well in solving the inverse problem.

This is similar results as for unsupervised inverse problems training loss.
[Tachella et al., 2022]

Recovery of a D generated as a 100×100 normalized Gaussian dictionary

A_i is a random $m \times 100$ sensing matrix, with Bernouilli-gaussian signals z_i



Convolutional Priors in Inverse Problems

Convolutional structures are widely used as priors for inverse problems.

We can consider Convolutional Dictionary Learning as a simple model of convolutional priors:

$$x^*(y; D) = D * \operatorname*{argmin}_z \frac{1}{2}\|y - A(D * z)\|_2^2 + \lambda\|z\|_1$$

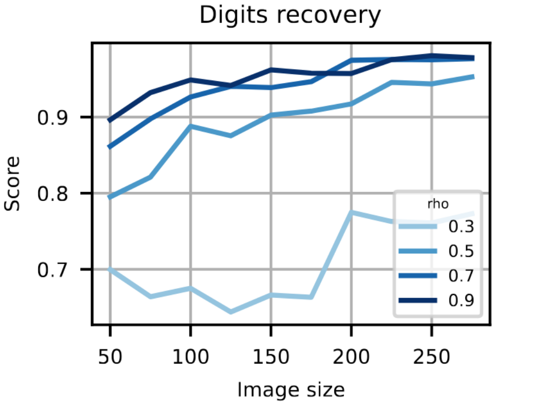
- Convolutional structure encodes translation invariance.
- The dictionary atoms capture local spatial patterns.
- Constrain local context to be similar to learned filters.

$\Rightarrow$ But how does this structure impact different inverse problems?

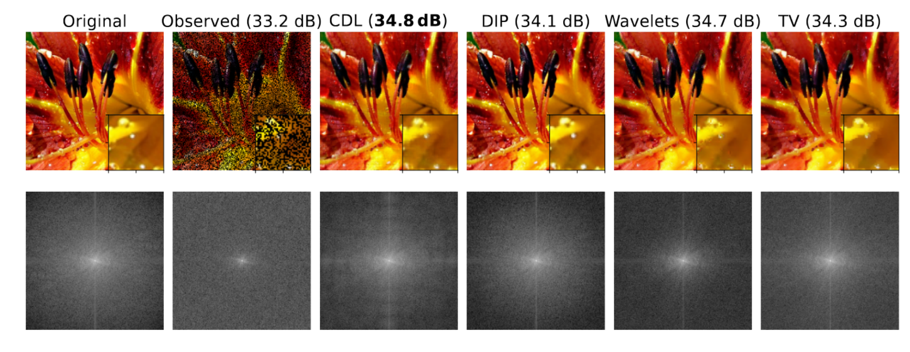
Inpainting: Leveraging Local Structure

Inpainting involves recovering missing pixels using surrounding information.

If the signal is stationary, convolutional prior on a single large measurement acts as on multiple measurements for each patch independently.



Inpainting: Leveraging Local Structure

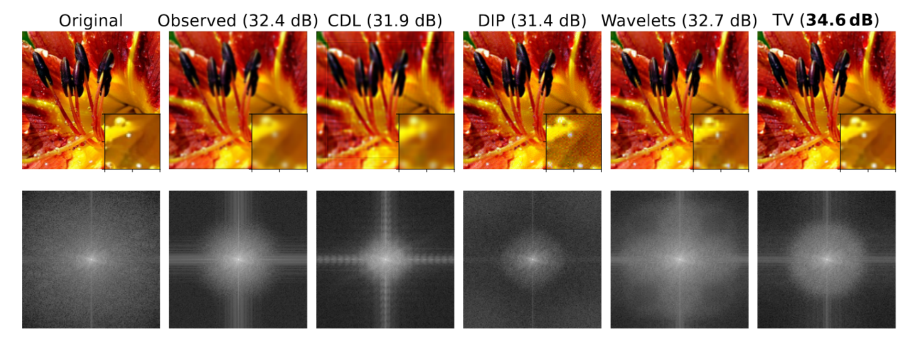


This structure is well adapted as it captures the distribution of local patterns, which is sufficient to fill in missing pixels.

Deblurring: The Challenge for Convolutional Priors

Deblurring aims to recover sharp images from blurred observations.

In this case, the convolutional structure is less adapted, as the kernel of the blur is aligned with the spectral structure of the prior.



For each patch in the image, the kernel is the same, so the convolutional structure does not provide the same advantage as in inpainting.

Takeaways on prior learning

- Dictionary learning is a useful tool to study prior learning.
- Understanding the structure of the kernel of $\boldsymbol{A}$ is crucial to design or learn a good prior.
- Our aim should be to find priors that links $P_{\text{Ker}\boldsymbol{A}}(\boldsymbol{x})$ to the rest of the image.

$\Rightarrow$ Evaluating priors based on this criterion should help select better priors.

This is typically what is done with splitting loss or equivariant learning.

FiRe: Fixed-Point Restoration

References

- Terris, M., Kamilov, U., and **TM** (2025). FiRe: Fixed-points of Restoration Priors for Solving Inverse Problems. In *CVPR*

Using restoration networks as priors

Many efficient restoration networks have been proposed for various tasks in the last years.

- JPEG restoration: SCUNet [Zhang et al., 2023]
- Deblurring: Restormer [Zamir et al., 2022]
- Inpainting: LAMA [Suvorov et al., 2022]
- ...

These models are trained to solve $p(\boldsymbol{x}|\boldsymbol{y})$ for a specific degradation $\boldsymbol{y} = D(\boldsymbol{x})$.

A good network for deblurring should be able to recover high frequencies from a blurry image.

$\Rightarrow$ Can we use them to solve other inverse problems?

TL;DR: *yes, this is what is done with DRP, SHARP, ...*
[Hu et al., 2024a, 2024b]

Fixed-point of restoration networks

Observation: Denoisers are not *stable* when iterated

$X_k = \underbrace{D \circ D \cdots \circ D(X_0)}_{k \text{ times}}$ is not converging to a realistic image

Here with **DRUNet** with $\sigma = 0.05$ [Zhang et al., 2021]

$\boldsymbol{x}_0$

$\boldsymbol{x}_1$

$\boldsymbol{x}_3$

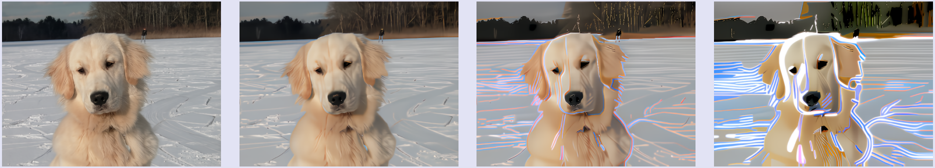
$\boldsymbol{x}_{10}$

Fixed-point of restoration networks

Observation: Similar observation holds for restoration networks

Here with **SCUNet** [Zhang et al., 2023]

x_0 x_1 x_{20} x_{49}

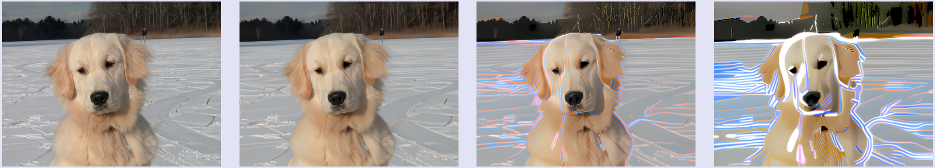


Fixed-point of restoration networks

Observation: Similar observation holds for restoration networks

Here with **SCUNet** [Zhang et al., 2023]

x_0 x_1 x_{20} x_{49}



and **Restformer** [Zamir et al., 2022]

Definition

A restoration model R^D adapted to a degradation D is model solving

$$\mathrm{R}^D(D(\boldsymbol{x})) \approx \boldsymbol{x} \quad \text{for images} \quad \boldsymbol{x} \sim \mathcal{X} \ .$$

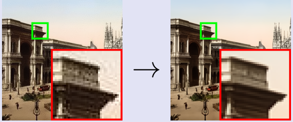
We have in mind degradation models $D(x)$ of the form

$$D(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + w,$$

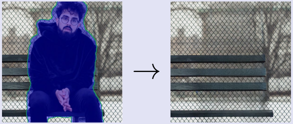
with $\boldsymbol{A}$ a linear operator, and some noise $w \sim \mathcal{W}$.

Restormer
Deblurring

SCUNet
JPEG Restoration



LAMA
inpainting



Training of restoration models

Direct restoration models are trained in a supervised manner, starting from clean images $x \sim \mathcal{X}$, and a degradation model D.

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{x}, w \sim \mathcal{W}} \left[\| \mathrm{R}_\theta^D (H\boldsymbol{x} + w) - x \| \right].$$

Observation:

- Let $T = \mathrm{R}^D \circ D$, then for R^D sufficiently well trained, we expect $T^2 \approx T$.
- T shows an idempotence property so if fixed-points exist, they should be for $\mathrm{R} \circ D$ and not R.
- With this training, we expect that the original data $\boldsymbol{x}$ are part of the fixed-point set of T.

Fixed-point of restoration networks

Observation: degradation + denoisers are *stable* when iterated

$X_k = \underbrace{D \circ D \cdots \circ D(X_0)}_{k \text{ times}}$ is not converging to a realistic image

Here with **DRUNet** with $\sigma = 0.05$ [Zhang et al., 2021]

$\mathbf{x}_0$ $\mathbf{x}_1$ $\mathbf{x}_3$ $\mathbf{x}_{10}$

and for T:

Observation: Similar observation holds for restoration networks

x_0

SCUNet

Restformer

and for T:

Restoration models as projections

Assumption

Let $C = \{x \in \mathbb{R}^n; T(x) = x\}$, then $T = \mathrm{R} \circ D$ can be expressed as a projection $T = \mathrm{proj}_C$ onto a closed, prox-regular set C.

Proposition

Under our projection assumption, around any point $\boldsymbol{x}$ where C is prox-regular, we have:

$$T(\boldsymbol{x}) = \boldsymbol{x} - \frac{1}{2}\nabla d_C^2(\boldsymbol{x}),$$

where: $d_C(\boldsymbol{x}) = \inf_{u \in C} \|\boldsymbol{x} - u\|$ is the distance to the set C.

$\Rightarrow$ Define a prior $p(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2} d_C^2(\boldsymbol{x})\right)$, promoting fixed points of T.

A anomaly detection view on restoration networks

T is a reconstruction network, that can be viewed as an auto-encoder with a fixed encoder D.

Anomaly detection literature: [Liu and Paparrizos 2024]

- Train a reconstruction network on normal data,
- use large reconstruction error as an anomaly indicator.

$$\text{Anomaly score}(\boldsymbol{x}) = \|\boldsymbol{x} - T(\boldsymbol{x})\|^2 = d_C^2(\boldsymbol{x})$$

$\Rightarrow$ Characterize $p(\boldsymbol{x}|D(\boldsymbol{x}))$ for $\boldsymbol{x} \sim \mathcal{X}$ a natural image.

This is usually easier to learn than $p(\boldsymbol{x})$ directly.

FiRe-HQS Algorithm

We can define the FiRe-HQS algorithm as:

$$u_k = x_k - \frac{\gamma}{2}\nabla d_C^2(x)$$
$$x_{k+1} = \text{prox}_{\lambda f}(u_k).$$

Note: This is also similar to SNORE for denoisers. [Renaud et al. 2024] .

Proposition

Under our projection assumption, the FiRe-HQS algorithm converges to a point x^* satisfying

$$x^* = \underset{x}{\text{argmin}}\ \lambda f(x) + \frac{\gamma}{2} d_C^2(x),$$

Extended FiRe-HQS Algorithm

We can extend the FiRe-HQS algorithm to multiple restoration models $\{T_i = \mathrm{R}_i \circ D_i\}_{i=1}^M$ as:

Input: Initial estimate x_0, weights γ_n, regularization parameter λ.
for $k = 1, \dots, K$ **do**
  for $n = 1, \dots, N$ **do**
    Select restoration model (R^n, D^n);
    Compute residual: $r_k^n = x_k - \mathrm{R}^n(D^n(x_k))$;
  $u_k = x_k - \sum_{n=1}^N \gamma_n r_k^n \quad \leftarrow \quad x_k - \gamma \nabla d_C^2(x_k)$;
  $x_{k+1} = \mathrm{prox}_{\lambda f}(u_k)$;
Output: Final estimate x_{k+1}.

$\Rightarrow$ We combine the strengths of multiple restoration models to remove artifacts from each other.

Key point: They are all trained to have the natural images as fixed-points.

Experimental Setting

We consider inverse problems $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \epsilon$ and solve:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \lambda \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|}_{f(\boldsymbol{x})} + \frac{\gamma}{2}\mathbb{E}_{\xi\sim\Xi}\left[d^2_{C_\xi}(\boldsymbol{x})\right]$$

for various implicit priors (recall that $\frac{1}{2}\nabla d_C(\boldsymbol{x}) = \boldsymbol{x} - \mathrm{R}(H\boldsymbol{x} + w)$).

Restoration Models R:

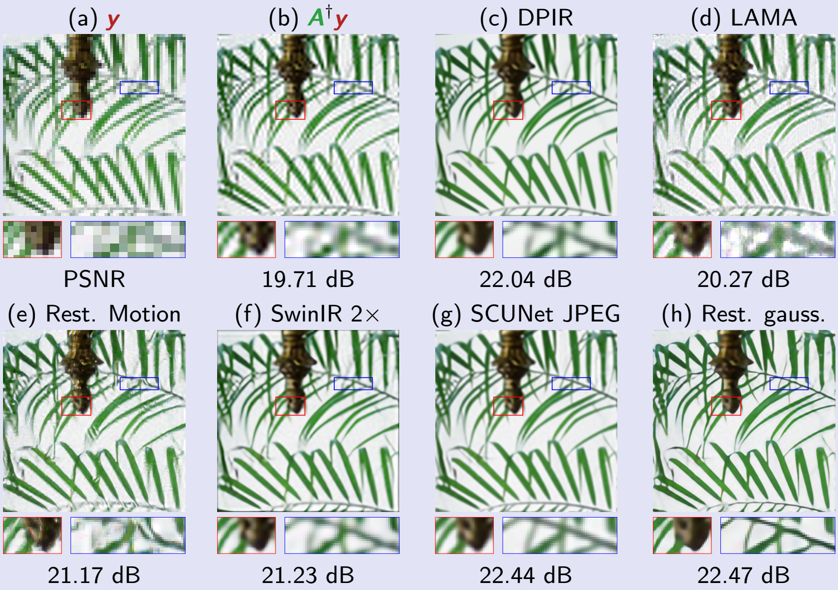
- **DRUNet:** Gaussian denoising with $H = \mathrm{Id}$ and $w \sim \mathcal{N}(0, \sigma^2)$.
- **Restormer:** Gaussian or motion deblurring with H as Gaussian or motion blurs, $w \sim \mathcal{N}(0, \sigma^2)$.
- **SCUNet:** Non-linear restoration with $H = \mathrm{JPEG}_q$ ($q \in [20, 100]$), $w \sim \mathcal{N}(0, \sigma^2)$.
- **SwinIR:** Super-resolution ($\times 2, \times 3$) with H as downsampling, $w = 0$.
- **LAMA:** (a) Pretrained: H as a large mask, $w = 0$. (b) Fine-tuned: H for random inpainting, $w = 0$.

Results with a single prior

We first consider our FiRe approach with a **single** restoration model. We consider the $4\times$ SR problem.

Case 1: prior is the SCUNet with noisy JPEG degradations.

Results with a single prior

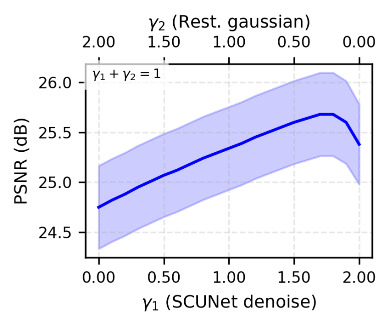


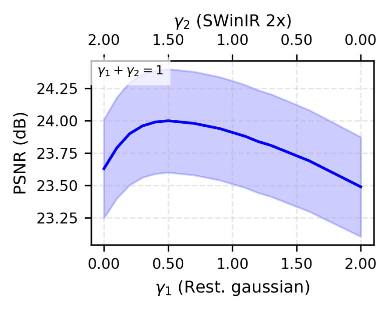
Combining priors

The FiRe framework allows us to combine priors as

$$\gamma_1 R_1(H_1 \boldsymbol{x} + w_1) + \gamma_2 R_2(H_2 \boldsymbol{x} + w_2)$$

Example:





γ vs PSNR within the reconstruction quality for two different problems. Left: Gaussian deblurring, right: SR$\times$4. The γ_1 and γ_2 parameter control the strength of the associated prior.

Visual results:

Observed	DRP	DPIR	DiffPIR	Proposed	Groundtruth
	(22.02, 0.52)	(24.02, 0.55)	(24.72, 0.49)	(25.11, 0.44)	(PSNR, LPIPS)
	(22.68, 0.27)	(29.19, 0.11)	(29.42, 0.09)	(28.64, 0.10)	(PSNR, LPIPS)
	(30.22, 0.22)	(30.03, 0.32)	(30.42, 0.27)	(30.17, 0.31)	(PSNR, LPIPS)

Image restoration with various algorithms. Top: SR$\times$4 problem with $\sigma = 0.01$ on BSD20. Middle: Motion blur on Imnet100. Bottom: Gaussian deblurring with blur kernel of size 3 and $\sigma = 0.01$ on Imnet100.

Conditioning the prior on the measurements

Recall the iteration (simplified)

$$u_k = R(H\mathbf{x}_k + w_k)$$

$$\mathbf{x}_{k+1} = \text{prox}_{\lambda f}(u_k).$$

Given that $f = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2$, one can set $H = \mathbf{A}$. Application to inpainting:

(a) Observed | (b) Denoise First | (c) Inpaint First | (d) Denoising Prior | (e) Inpainting Prior

(f) DiffPIR | (g) DRP | (h) ShaRP | (i) **Proposed**

Conclusion

Conclusion and perspectives

- Priors structure should be adapted to the task and data
- Restoration networks can be used to define priors promoting natural images as fixed-points
- FiRe-HQS is a flexible algorithm leveraging multiple restoration models to solve inverse problems

Reproducible method comparison with Benchopt



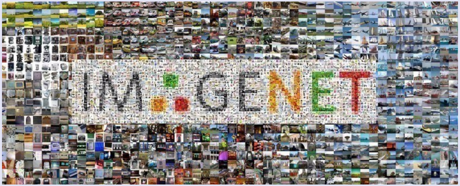



. . .

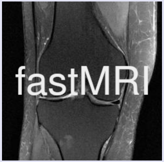
References

- **TM**, Massias, M., Gramfort, A., Ablin, P., Bannier, P.-A., Charlier, B., Dagréou, M., la Tour, T. D., Durif, G., Dantas, C. F., Klopfenstein, Q., Larsson, J., Lai, E., Lefort, T., Malézieux, B., Moufad, B., Nguyen, B. T., Rakotomamonjy, A., Ramzi, Z., Salmon, J., and Vaiter, S. (2022). Benchopt: Reproducible, efficient and collaborative optimization benchmarks. In *NeurIPS*

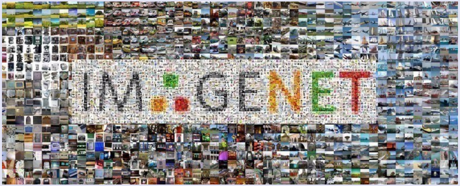
Benchmarks fueled AI progress



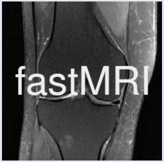
GLUE



Benchmarks fueled AI progress



GLUE



But sometime, it is not so clear which methods should be included:

- Different evaluation protocols
- Different implementations
- Hard to tune all methods
- ...

$\Rightarrow$ Many novel methods but unclear improvements

benchopt provides a framework to organize and run benchmarks

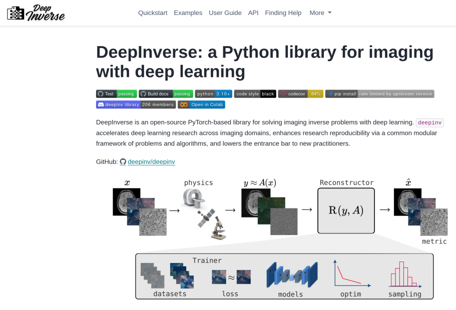
Examples of existing benchmarks:

- **Image Classification (resnet)**
- **Logistic regression**
- **Lasso**
- **ICA**
- **Unsup. Domain Adaptation**
- **Bilevel Optimization**
- **Brain Computer Interface**
- ...

Example: Benchmarking Inverse Problems solvers with Deepinv

Benchmarking various methods in a single repo:

- Imaging, MRI, CT, . . .
- Direct, PnP, Variational, . . .
- Centralized evaluation
- Clear rules on tuning the methods



⇒ **Goal:** Make it easy to add new methods and datasets

If you are interested, happy to discuss this week!

https://github.com/benchopt/benchmark_inverse_problems/