Some thoughts about PnP, unrolled and diffusion

Pierre Weiss, IRIT & CBI, Université de Toulouse

Why I came today?

Why I came today?

A mental disorder started in 2022...

- Workshop, Mathematical models for **PnP** restoration, *MIA* 2022
- Laumont, Bayesian computation with **PnP** priors for inverse problems, *PhD* 2022
- Hurault, Convergent **PnP** methods for inverse problems, *PhD* 2023
- Gossard & P.W., Adaptive **unrolled networks** for blind inverse problems, *SIIMS* 2022

Why I came today?

A mental disorder started in 2022...

- Workshop, Mathematical models for **PnP** restoration, *MIA* 2022
- Laumont, Bayesian computation with **PnP** priors for inverse problems, *PhD* 2022
- Hurault, Convergent **PnP** methods for inverse problems, *PhD* 2023
- Gossard & P.W., Adaptive **unrolled networks** for blind inverse problems, *SIIMS* 2022

Am I an outdated marginal?

Why I came today?

A mental disorder started in 2022...

- Workshop, Mathematical models for **PnP** restoration, *MIA* 2022
- Laumont, Bayesian computation with **PnP** priors for inverse problems, *PhD* 2022
- Hurault, Convergent **PnP** methods for inverse problems, *PhD* 2023
- Gossard & P.W., Adaptive **unrolled networks** for blind inverse problems, *SIIMS* 2022

Am I an outdated marginal?
Objective today: clarify this!

An outdated marginal?

Maybe not... Own experiments suggest that unrolled nets are “better”

Groud truth

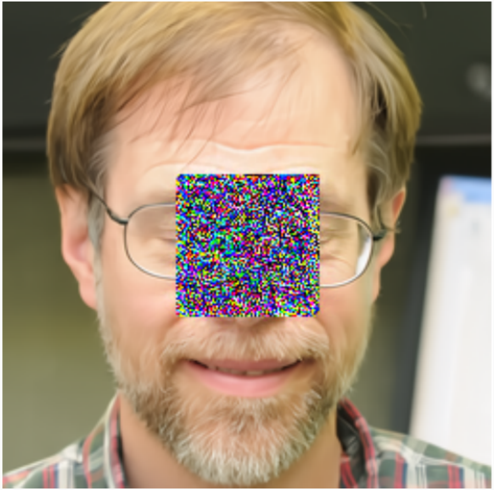
An outdated marginal?

Maybe not... Own experiments suggest that unrolled nets are “better”

Masked image

An outdated marginal?

Maybe not... Own experiments suggest that unrolled nets are "better"



Vanilla PnP (DRUNet)

An outdated marginal?

Maybe not... Own experiments suggest that unrolled nets are "better"

Vanilla Unrolled (same architecture)

Maybe not... Own experiments suggest that unrolled nets are "better"

Diffusion model (advanced PnP)

Why I came today?

PnP and unrolled, what is different?

- Definitions
- Statistical interpretation
- Qualitative properties
- Some insights

Inverse problems

$$y = A(x) + b$$

- $A : \mathbb{R}^N \to \mathbb{R}^M$ observation operator
- x: object to recover (may be more complicated than an image)
- b: noise
- y: observed measurements

Inverse problem $\equiv$ recover x from y

Bayesian formalism

Some information is lost in the acquisition!

We inject it through a probabilistic model.

- x is the realization of a random variable X with density p_X.
- b is the realization of a random variable B with density p_B.

Popular estimators

MAP = best point estimate

Maximum A Posteriori (optimization):

$$\begin{aligned}\hat{x}_{MAP}(y) &\overset{\text{def.}}{=} \operatorname*{argmax}_x p_{X|Y}(x|y) \\ &\overset{Bayes}{=} \operatorname*{argmin}_x -\log p_{Y|X}(y|x) - \log(p_X(x)) \\ &= \operatorname*{argmin}_x f(x) + g(x).\end{aligned}$$

Example: $b \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}) \Rightarrow f(x) = \frac{1}{2\sigma^2} \|A(x) - y\|_2^2$

Popular estimators

MAP = best point estimate

Maximum A Posteriori (optimization):

$$\begin{aligned}\hat{x}_{MAP}(y) &\overset{\text{def.}}{=} \operatorname*{argmax}_x p_{X|Y}(x|y) \\ &\overset{Bayes}{=} \operatorname*{argmin}_x -\log p_{Y|X}(y|x) - \log(p_X(x)) \\ &= \operatorname*{argmin}_x f(x) + g(x).\end{aligned}$$

Example: $b \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}) \Rightarrow f(x) = \frac{1}{2\sigma^2}\|A(x) - y\|_2^2$

MMSE = best in average estimate

Minimum Mean Square Estimation (integration):

$$\begin{aligned}\hat{x}_{MMSE}(y) &\overset{\text{def.}}{=} \operatorname*{argmin}_{x \in \mathbb{R}^N} \mathbb{E}(\|x - X\|_2^2 | Y = y) = \mathbb{E}(x|y) \\ &= \int_x x \cdot dp_{x|y}(x)\end{aligned}$$

How to compute them?

Popular estimators

MAP = best point estimate

Maximum A Posteriori (optimization):

$$\begin{aligned}\hat{x}_{MAP}(y) &\overset{\text{def.}}{=} \underset{x}{\operatorname{argmax}}\, p_{X|Y}(x|y) \\ &\overset{Bayes}{=} \underset{x}{\operatorname{argmin}} - \log p_{Y|X}(y|x) - \log(p_X(x)) \\ &= \underset{x}{\operatorname{argmin}}\, f(x) + g(x).\end{aligned}$$

Example: $b \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}) \Rightarrow f(x) = \frac{1}{2\sigma^2} \|A(x) - y\|_2^2$

MMSE = best in average estimate

Minimum Mean Square Estimation (integration):

$$\begin{aligned}\hat{x}_{MMSE}(y) &\overset{\text{def.}}{=} \underset{x \in \mathbb{R}^N}{\operatorname{argmin}}\, \mathbb{E}(\|x - X\|_2^2 | Y = y) = \mathbb{E}(x|y) \\ &= \int_x x \cdot dp_{x|y}(x)\end{aligned}$$

Popular estimators

MAP = best point estimate

Maximum A Posteriori (optimization):

$$\begin{aligned}\hat{x}_{MAP}(y) &\overset{\text{def.}}{=} \underset{x}{\operatorname{argmax}}\, p_{X|Y}(x|y)\\ &\overset{Bayes}{=} \underset{x}{\operatorname{argmin}} -\log p_{Y|X}(y|x) - \log(p_X(x))\\ &= \underset{x}{\operatorname{argmin}}\, f(x) + g(x).\end{aligned}$$

Example: $b \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}) \Rightarrow f(x) = \frac{1}{2\sigma^2}\|A(x) - y\|_2^2$

MMSE = best in average estimate

Minimum Mean Square Estimation (integration):

$$\begin{aligned}\hat{x}_{MMSE}(y) &\overset{\text{def.}}{=} \underset{x \in \mathbb{R}^N}{\operatorname{argmin}}\, \mathbb{E}(\|x - X\|_2^2 | Y = y) = \mathbb{E}(x|y)\\ &= \int_x x \cdot dp_{x|y}(x)\end{aligned}$$

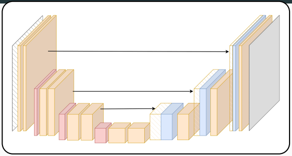
How to compute them?

MMSE and supervised learning

Computing the MMSE for $Y = A(X) + B$

MMSE and supervised learning

Computing the MMSE for $Y = A(X) + B$

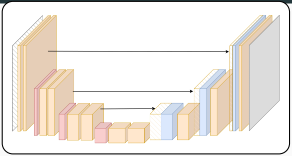


Prerequisities

- Neural network $N(y, w)$.
- A database of clean images $(x_1, \dots, x_I)$
- Synthesize $y_i = A(x_i) + b_i$.

MMSE and supervised learning

Computing the MMSE for $Y = A(X) + B$



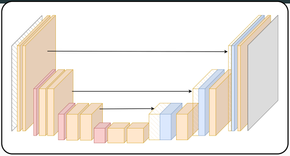
Prerequisities

- Neural network $N(y, w)$.
- A database of clean images $(x_1, \ldots, x_I)$
- Synthesize $y_i = A(x_i) + b_i$.

Training $\equiv$ Stochastic gradient

- $\inf_w \frac{1}{I} \sum_{i=1}^{I} \|N(y_i, w) - x_i\|_2^2$

MMSE and supervised learning



Computing the MMSE for $Y = A(X) + B$

Prerequisities

- Neural network $N(y, w)$.
- A database of clean images $(x_1, \ldots, x_I)$
- Synthesize $y_i = A(x_i) + b_i$.

Training $\equiv$ Stochastic gradient

- $\inf_w \frac{1}{I} \sum_{i=1}^{I} \|N(y_i, w) - x_i\|_2^2$

Output

- $N(y, w^\star)$: a trained network
- Can be used with arbitrary images

MMSE and supervised learning

Claim (informal)

If I large enough, $N(\cdot, w)$ expressive + generalizes, good training.

$$N(y, w^\star) \approx \hat{x}_{MMSE}(y)$$

MMSE and supervised learning

Claim (informal)

If I large enough, $N(\cdot, w)$ expressive + generalizes, good training.

$$N(y, w^\star) \approx \hat{x}_{MMSE}(y)$$

Proof.

$$\begin{aligned}
N(\cdot, w^\star) &\approx \underset{\phi = N(\cdot, w)}{\operatorname{argmin}} \frac{1}{I} \sum_{i=1}^{I} \|\phi(y_i) - x_i\|_2^2 && \text{Good optimization} \\
&\approx \underset{\phi \text{ measurable}}{\operatorname{argmin}} \frac{1}{I} \sum_{i=1}^{I} \|\phi(y_i) - x_i\|_2^2 && \text{Expressivity + Generalization} \\
&\approx \underset{\phi \text{ measurable}}{\operatorname{argmin}} \mathbb{E}_{X,Y}(\|\phi(Y) - X\|_2^2) && \text{Large dataset} \\
&\overset{\text{def.}}{=} \hat{x}_{MMSE}!
\end{aligned}$$

$\square$

MAP and Gradient methods (a 1 page panorama)

Combettes & Pesquet, Proximal splitting methods in signal processing, 2011

"Implicit" gradient

$$x_{k+1} = x_k - \tau \nabla f(x_{k+1}) \Leftrightarrow x_{k+1} = \mathrm{prox}_{\tau f}(x_k)$$

MAP and Gradient methods (a 1 page panorama)

Combettes & Pesquet, Proximal splitting methods in signal processing, 2011

"Implicit" gradient

$$x_{k+1} = x_k - \tau \nabla f(x_{k+1}) \Leftrightarrow x_{k+1} = \mathrm{prox}_{\tau f}(x_k)$$

Explicit/Implicit methods for $f + g$

- E-E (gradient descent):

$$x_{k+1} = x_k - \tau_k \left(\nabla f(x_k) + \nabla g(x_k) \right)$$

- I-E (proximal gradient descent):

$$x_{k+1} = \mathrm{prox}_{\tau_k f} \left(x_k - \tau_k \nabla g(x_k) \right)$$

- E-I (proximal gradient descent):

$$x_{k+1} = \mathrm{prox}_{\tau_k g} \left(x_k - \tau_k \nabla f(x_k) \right)$$

- I-I (Douglas-Rachford, ADMM): both prox_g and prox_f.

MAP: computing $\nabla g = -\nabla \log p_X$

Claim

Let D_δ denote a network trained for denoising

$$Y = X + B, \quad B \sim \mathcal{N}(0, \delta^2 \mathrm{Id})$$

Let $g_\delta = -\log(p_Y)$ Then

$$\nabla g_\delta(x) \approx \frac{x - D_\delta(x, w^\star)}{\delta^2}$$

Good denoiser $\approx$ gradient of the log prior!

MAP: computing $\nabla g = -\nabla \log p_X$

Claim

Let D_δ denote a network trained for denoising

$$Y = X + B, \quad B \sim \mathcal{N}(0, \delta^2 \mathrm{Id})$$

Let $g_\delta = -\log(p_Y)$ Then

$$\nabla g_\delta(x) \approx \frac{x - D_\delta(x, w^\star)}{\delta^2}$$

Good denoiser $\approx$ gradient of the log prior!

Proof.

$$p_Y = p_X \star G_\delta \qquad \text{Basic property}$$

MAP: computing $\nabla g = -\nabla \log p_X$

Claim

Let D_δ denote a network trained for denoising

$$Y = X + B, \quad B \sim \mathcal{N}(0, \delta^2 \mathrm{Id})$$

Let $g_\delta = -\log(p_Y)$ Then

$$\nabla g_\delta(x) \approx \frac{x - D_\delta(x, w^\star)}{\delta^2}$$

Good denoiser $\approx$ gradient of the log prior!

Proof.

$$p_Y = p_X \star G_\delta \qquad \text{Basic property}$$

$$\begin{aligned}
\nabla \log p_Y(y) &= \frac{y - \hat{x}_{MMSE}(y)}{\delta^2} && \text{Tweedie Formula} \\
&\approx \frac{y - D_\delta(y, w^\star)}{\delta^2} && \text{NN power} \\
&\approx \nabla \log p_X(y) \approx \nabla g(x) && \text{Small } \delta
\end{aligned}$$

MAP: computing $\mathrm{prox}_{\tau g}$

Claim: hardly tractable

$$\mathrm{prox}_{\tau g}(x_0) \overset{\text{def.}}{=} \operatorname*{argmin}_x -\tau \log p_X(x) + \frac{1}{2}\|x - x_0\|_2^2$$

Nonconvex, full of spurious minimizers for large τ.

MAP: computing $\mathrm{prox}_{\tau g}$

Claim: hardly tractable

$$\mathrm{prox}_{\tau g}(x_0) \overset{\text{def.}}{=} \operatorname*{argmin}_{x} -\tau \log p_X(x) + \frac{1}{2}\|x - x_0\|_2^2$$

Nonconvex, full of spurious minimizers for large τ.

A common practice

People replace $\mathrm{prox}_{\tau g}$ by $D_\delta(x_0)$.

MAP: computing $\mathrm{prox}_{\tau g}$

Claim: hardly tractable

$$\mathrm{prox}_{\tau g}(x_0) \stackrel{\text{def.}}{=} \operatorname*{argmin}_x -\tau \log p_X(x) + \frac{1}{2}\|x - x_0\|_2^2$$

Nonconvex, full of spurious minimizers for large τ.

A common practice

People replace $\mathrm{prox}_{\tau g}$ by $D_\delta(x_0)$.

Understanding still limited, though it was the original PnP

Computing the MAP - an example

Regularization by denoising

Assume that $Y = A(X) + B$.

$$\hat{x}_{MAP}(y) \overset{e.g.}{=} \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|Ax - y\|_2^2 - \log p_X(x)$$

Computing the MAP - an example

Regularization by denoising

Assume that $Y = A(X) + B$.

$$\hat{x}_{MAP}(y) \overset{e.g.}{=} \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|Ax - y\|_2^2 - \log p_X(x)$$

Can be computed with a gradient descent:

$$\begin{aligned} x_{k+1} &= x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \nabla \log p_X(x_k) \right] \\ &\overset{\delta \ll 1}{\approx} x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_\delta(x_k, w^\star)}{\delta^2} \right]. \end{aligned}$$

This is Vanilla PnP

Computing the MAP - an example

Regularization by denoising

Assume that $Y = A(X) + B$.

$$\hat{x}_{MAP}(y) \overset{e.g.}{=} \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|Ax - y\|_2^2 - \log p_X(x)$$

Can be computed with a gradient descent:

$$\begin{aligned} x_{k+1} &= x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \nabla \log p_X(x_k) \right] \\ &\overset{\delta \ll 1}{\approx} x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_\delta(x_k, w^\star)}{\delta^2} \right]. \end{aligned}$$

This is Vanilla PnP

Not working for highly ill-posed problems.

Assume that $Y = A(X) + B$. Construct the Euler-Maruyama sequence:

$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_\delta(x_k, w^\star)}{\delta^2} + \sqrt{2} b_k \right]$$

where $b_k \sim \mathcal{N}(0, \mathrm{Id})$.

Then (under mild conditions – log-Sobolev inequalities)

$$\frac{1}{K} \sum_{k=1}^{K} \delta_{x_k} \rightharpoonup p_{X|Y}$$

Sampling the posterior (Langevin diffusion)

Assume that $Y = A(X) + B$. Construct the Euler-Maruyama sequence:

$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_\delta(x_k, w^\star)}{\delta^2} + \sqrt{2} b_k \right]$$

where $b_k \sim \mathcal{N}(0, \mathrm{Id})$.

Then (under mild conditions – log-Sobolev inequalities)

$$\frac{1}{K} \sum_{k=1}^{K} \delta_{x_k} \rightharpoonup p_{X|Y}$$

My experience: not working for hard problems

Assume that $Y = A(X) + B$. Construct the Euler-Maruyama sequence:

$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_{\delta_k}(x_k, w^\star)}{\delta^2} + \left(\sqrt{2} b_k \right) \right].$$

with $\delta_0 \geq \delta_1 \geq \ldots \geq \delta_K \approx 0$.

Diffusion / Gaussian continuation / score matching

Assume that $Y = A(X) + B$. Construct the Euler-Maruyama sequence:

$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(y|x_k) - \frac{x_k - D_{\delta_k}(x_k, w^\star)}{\delta^2} + \left(\sqrt{2} b_k\right) \right].$$

with $\delta_0 \geq \delta_1 \geq \ldots \geq \delta_K \approx 0$.

My experience: works, basic mechanism behind "diffusion" models

Computing the MMSE with unrolled networks

Assume that $Y = A(X) + B$.

Construct a sequence of denoising networks $D(x, w_k)$, $k = 1 \dots K$:

$$x_0 = A^{-1}(y)$$
$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(x_k|y) - \frac{x_k - D(x_k, w_k)}{\delta^2} \right]$$

Computing the MMSE with unrolled networks

Assume that $Y = A(X) + B$.

Construct a sequence of denoising networks $D(x, w_k)$, $k = 1 \dots K$:

$$x_0 = A^{-1}(y)$$
$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(x_k|y) - \frac{x_k - D(x_k, w_k)}{\delta^2} \right]$$

Define the architecture $\mathcal{UN}(y, A, w) = x_K$ with $w = (w_1, \dots, w_K)$.

After training:

$$\mathcal{UN}(y, A, w^\star) \approx \hat{x}_{MMSE}(y)!$$

Computing the MMSE with unrolled networks

Assume that $Y = A(X) + B$.

Construct a sequence of denoising networks $D(x, w_k)$, $k = 1 \dots K$:

$$x_0 = A^{-1}(y)$$

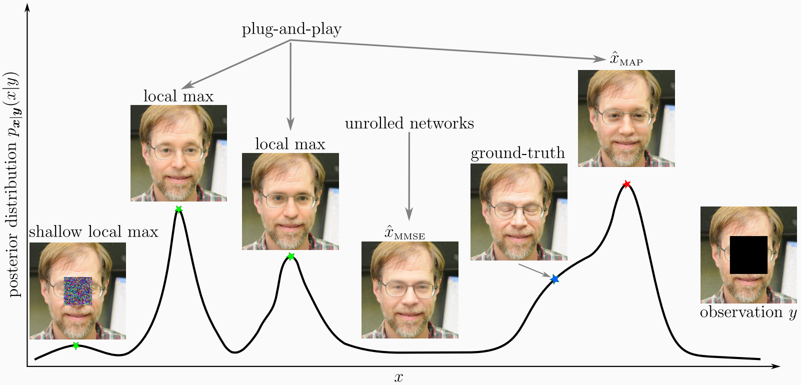
$$x_{k+1} = x_k - \tau \left[-\nabla \log p_{Y|X}(x_k|y) - \frac{x_k - D(x_k, w_k)}{\delta^2} \right]$$

Define the architecture $\mathcal{UN}(y, A, w) = x_K$ with $w = (w_1, \dots, w_K)$.

After training:

$$\mathcal{UN}(y, A, w^\star) \approx \hat{x}_{MMSE}(y)!$$

Unrolling = good for expressivity + generalization



plug-and-play
$\hat{x}_{\mathrm{MAP}}$
local max
unrolled networks
local max
ground-truth
shallow local max
$\hat{x}_{\mathrm{MMSE}}$
observation y
posterior distribution $p_{\boldsymbol{x}|\boldsymbol{y}}(x|y)$
x

Main facts

- MAP estimation
  - Learn to denoise $\approx$ prior via $\nabla \log p_X$
  - Universal: can be used for arbitrary inverse problems
  - Plug&play (universal method)
  - Vanilla not satisfactory
  - Requires continuation
  - Possibility to include sampling
  - "Best" looking result... But, can we trust it?
  - Can be slow at runtime

Main facts

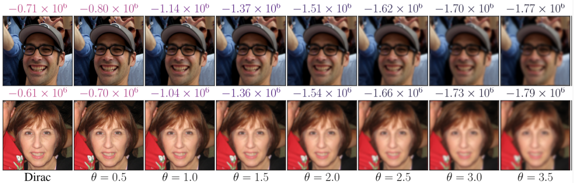
- MAP estimation
  - Learn to denoise $\approx$ prior via $\nabla \log p_X$
  - Universal: can be used for arbitrary inverse problems
  - Plug&play (universal method)
  - Vanilla not satisfactory
  - Requires continuation
  - Possibility to include sampling
  - "Best" looking result... But, can we trust it?
  - Can be slow at runtime

- MMSE
  - Unrolled network (specific to an operator)
  - Same architecture as PnP!
  - Learn to reconstruct
  - "Best" result in average (blurry where unfaithful)
  - Fast at runtime
  - Long at train time

	PnP methods	Unrolled networks
Stat. interpretation	MAP (local.)	MMSE
Architecture	Identical, but smaller K for unrolled	
Training objective	Learn to denoise	Learn to reconstruct
Training cost	Rather lightweight	Rather expensive
Adaptivity	Any inverse problem	Problem-dependent
Inference time	Long if many iterations	Fast once trained
Convergence	Local minimizers	Ongoing research
Computation	High dim. optimization	High dim. integral
Stability	Low for nonsmooth priors	More stable
Appearance	Best looking	Best in average
Performance	Unrolled $>$ vanilla-PnP by up to 5 dB [35]	
Properties	Nice looking but uncertain	Blurry where unfaithful
	Trapped in local minimizers	Can be unlikely
	Improved with continuation	

MAP = intrisically bad idea for blind deblurring

Recover θ, x from $y = h(\theta) \star x$



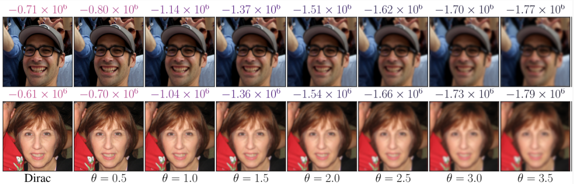
Theorem (Nguyen, Pauwels, P.W. 2024)

If p_X prior learned on natural images and $x \in \min_{local}(p_X)$. Then $p_{(\theta,x)|y}$ possesses:

- A global minimizer at (δ, y).

MAP = intrisically bad idea for blind deblurring

Recover θ, x from $y = h(\theta) \star x$



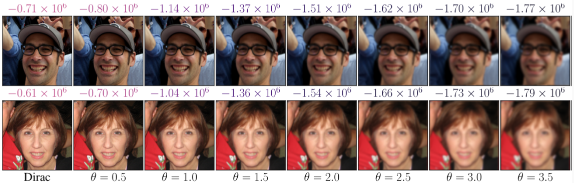
Theorem (Nguyen, Pauwels, P.W. 2024)

If p_X prior learned on natural images and $x \in \min_{local}(p_X)$. Then $p_{(\theta,x)|y}$ possesses:

- A global minimizer at (δ, y).
- A local minimizer at $(h(\theta), x)$.

MAP = intrisically bad idea for blind deblurring

Recover θ, x from $y = h(\theta) \star x$



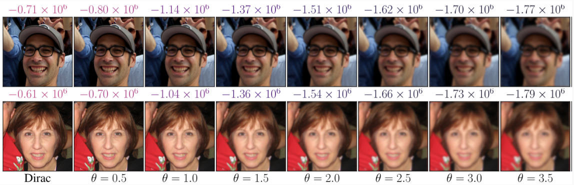
Theorem (Nguyen, Pauwels, P.W. 2024)

If p_X prior learned on natural images and $x \in \min_{local}(p_X)$. Then $p_{(\theta,x)|y}$ possesses:

- A global minimizer at (δ, y).
- A local minimizer at $(h(\theta), x)$.
- The result is stable to noise.

MAP = intrisically bad idea for blind deblurring

Recover θ, x from $y = h(\theta) \star x$



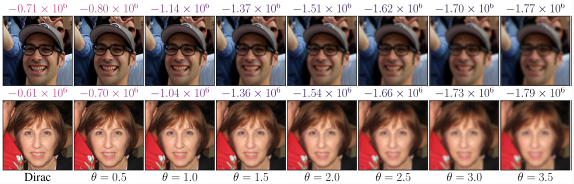
Theorem (Nguyen, Pauwels, P.W. 2024)

If p_X prior learned on natural images and $x \in \min_{local}(p_X)$. Then $p_{(\theta,x)|y}$ possesses:

- A global minimizer at (δ, y).
- A local minimizer at $(h(\theta), x)$.
- The result is stable to noise.
- Finding local minimizers is not tractable.

MAP = intrisically bad idea for blind deblurring

Recover θ, x from $y = h(\theta) \star x$



Theorem (Nguyen, Pauwels, P.W. 2024)

If p_X prior learned on natural images and $x \in \min_{local}(p_X)$. Then $p_{(\theta,x)|y}$ possesses:

- A global minimizer at (δ, y).
- A local minimizer at $(h(\theta), x)$.
- The result is stable to noise.
- Finding local minimizers is not tractable.

The MMSE behaves differently.

Conclusion

Maximizing the posterior can be risky

PnP, diffusion designed to find posterior maxima

- Full of spurious minimizers
- Partial avoidance with Gaussian continuation
- Diffusion can help, but still looking for modes
- The global maximizer can be pointless

Conclusion

Maximizing the posterior can be risky

PnP, diffusion designed to find posterior maxima

- Full of spurious minimizers
- Partial avoidance with Gaussian continuation
- Diffusion can help, but still looking for modes
- The global maximizer can be pointless

Unrolled networks = mutiple advantages

- Fast at inference time
- Blur = kind of uncertainty quantification.
- Stable
- Works empirically for blind inverse problems

A few personal references

More details

- Nguyen & P.W., Comparing PnP and Unrolled networks, *preprint* 2024
- Gossard & P.W., Training adaptive reconstruction networks for blind inverse problems, *SIAM Imaging Science* 2024
- Debarnot & P.W., DEEP-BLUR: Blind Identification and Deblurring with CNN, *Biological Imaging*, 2024
- Nguyen, Pauwels & P.W., How learned priors shape the posterior landscapes in blind inverse problems, *Preprint*, 2024