Variational Bayes image restoration for satellite imaging

Comet TSI

Maud Biquard[1,3]

Advisors: Thomas Oberlin[1], Marie Chabert[2], Florence Genin[3], Christophe Latry[3]

[maud.biquard@isae-supaero.fr]

26 juin 2024

[1] ISAE-SUPAERO, [2] IRIT [3] CNES

[*Intro*] Satellite image restoration

Noise sources

- Sensors default, photonic noise

Blur sources

- Optical system, satellite movement, atmosphere

Figure: Acquisition of the satellite and restored image

The optical system can be modeled as:

$$y = h * x + b \text{ where } b \sim \mathcal{N}(0, \sigma^2), \sigma^2 = a + bx$$

$$h = h_{atmo} * h_{filé} * h_{opt}$$

[*Intro*] Content

[*Method*] Neural networks based image restoration methods

Inverse problem: $y = Ax + n$ with n some noise.
Classical inversion process: $\arg\min_x ||y - Ax||^2 + \lambda R(x)$

[*Method*] Neural networks based image restoration methods

Inverse problem: $y = Ax + n$ with n some noise.
Classical inversion process: $\arg\min_x ||y - Ax||^2 + \lambda R(x)$

Direct inversion of the degradation:

- Supervised learning of a function $f : Y \rightarrow X$ from samples (x_i, y_i)
- Specific to the problem $\Rightarrow$ a network for each inverse problem

[Method] Neural networks based image restoration methods

Inverse problem: $y = Ax + n$ with n some noise.
Classical inversion process: $\arg\min_x ||y - Ax||^2 + \lambda R(x)$

Direct inversion of the degradation:

- Supervised learning of a function $f : Y \to X$ from samples (x_i, y_i)
- Specific to the problem $\Rightarrow$ a network for each inverse problem

Learning only the regularisation from (x_i):

- No need of simulated images (y_i)
- One network can be used to solve multiple inverse problems.

[*Method*] Neural networks based image restoration methods

Inverse problem: $y = Ax + n$ with n some noise.
Classical inversion process: $\arg\min_x ||y - Ax||^2 + \lambda R(x)$

Direct inversion of the degradation:

- Supervised learning of a function $f : Y \to X$ from samples (x_i, y_i)
- Specific to the problem $\Rightarrow$ a network for each inverse problem

Learning only the regularisation from (x_i):

- No need of simulated images (y_i)
- One network can be used to solve multiple inverse problems.

Using denoisers | *Using generative models*

[*Method*] Generative neural networks

- To synthesize realistic data
- Different types : GAN, VAE, Normalizing Flows, Diffusion Models
- Function G that maps $z \sim p_Z$ to $x = G(z)$ from the image distribution p_X

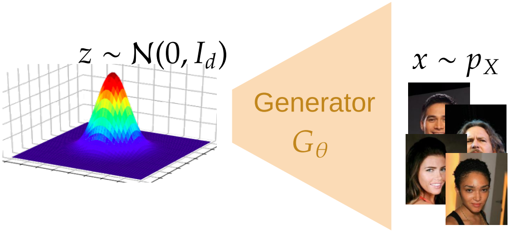


Figure: A generative model

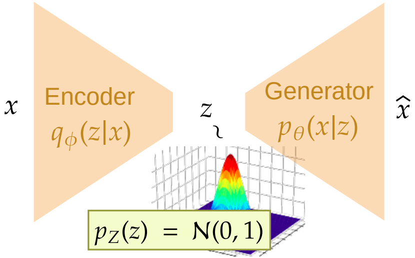


Figure: A variational autoencoder (VAE)

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

[1] A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.

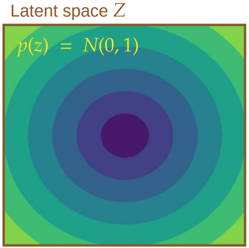


Figure: Restoration process of Bora's method[1]

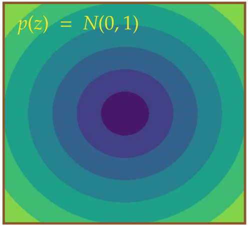
[1] A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.

Latent space Z



Measure y

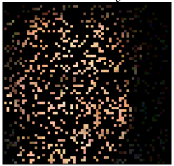


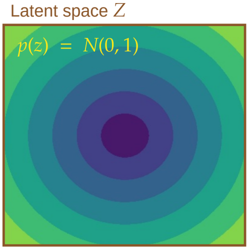
Figure: Restoration process of Bora's method[1]

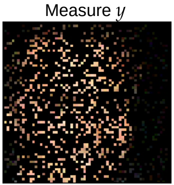
[1]A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.





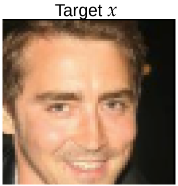


Figure: Restoration process of Bora's method[1]

[1] A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.

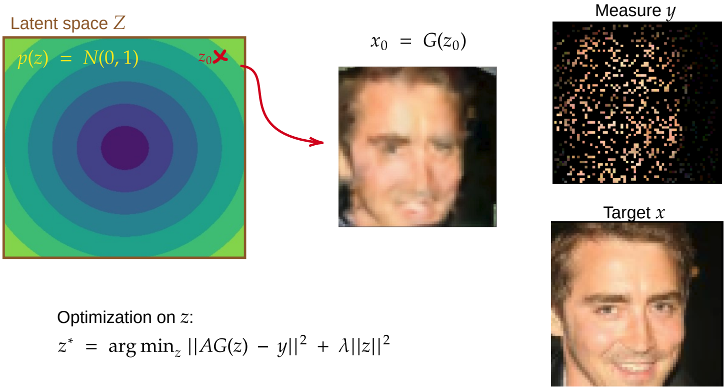


Figure: Restoration process of Bora's method[1]

[1]A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.

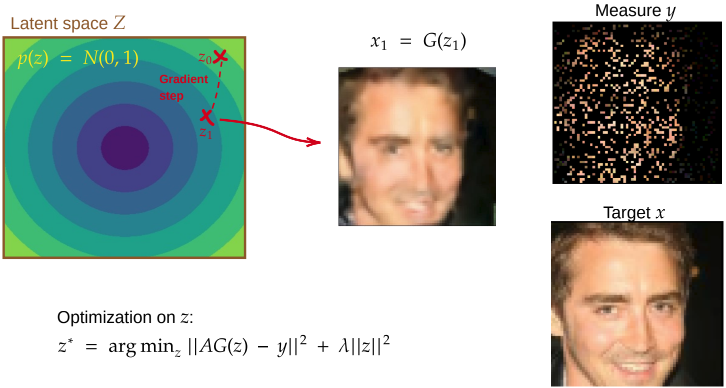


Figure: Restoration process of Bora's method[1]

[1] A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Compressed sensing using Generative Models[1]

First step: Train a generative model G on ideal images.

Second step: Looking for the solution in the latent space of G.

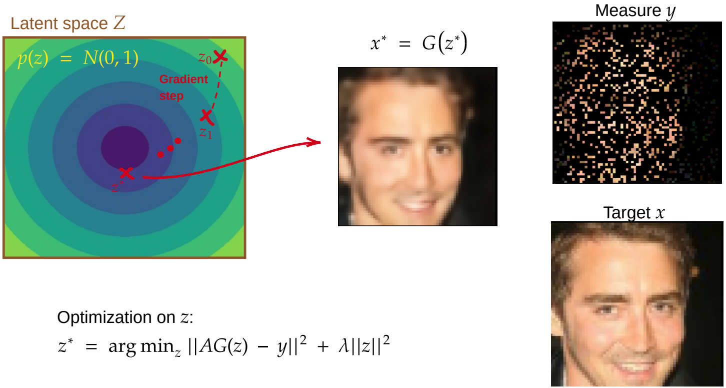


Figure: Restoration process of Bora's method[1]

[1] A.Bora, A.Jalal, E.Price, A.G.Dimakis, *Compressed Sensing using Generative Models*, ICML, 2017.

[*Method*] Variational Bayes Latent Estimation[1] (VBLE)

Contributions

- Utilisation d'une structure particulière d'autoencodeur génératif, muni d'un *hyperprior* flexible
- Proposition d'un algorithme dérivé qui permet de générer plusieurs solutions / estimer les incertitudes d'une solution

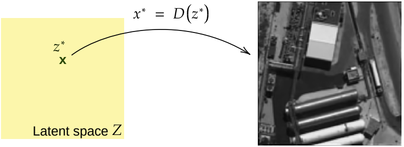


Figure: Latent MAP estimation

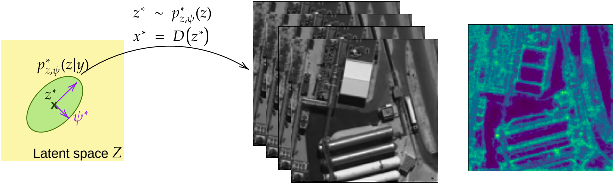


Figure: VBLE, enabling posterior sampling

[1] M. Biquard, M. Chabert, F. Genin, C. Latry, T. Oberlin, *Variational Bayes Image Restoration with compressive autoencoders*, 2024 (preprint)

[*Method*] Variational Bayes Latent Estimation

Formulation with variational inference

- Suppose $p_{Y|Z}(y|z) = p_{Y|X}(y|x = D_\theta(z))$.

[*Method*] Variational Bayes Latent Estimation

Formulation with variational inference

- Suppose $p_{Y|Z}(y|z) = p_{Y|X}(y|x = D_\theta(z))$.
- Approximation of the measure posterior $p_{Z|Y}(z|y)$ with $q_{\bar{z},a}(z)$ with

$$\bar{z}, a \in \mathbb{R}^{C\times M\times N}, a > 0 \text{ and } q_{\bar{z},a}(z) = \prod_k \mathcal{U}(z_k; [\bar{z}_k - \frac{a_k}{2}, \bar{z}_k + \frac{a_k}{2}]) \quad \text{(CAE case)},$$

$$q_{\bar{z},a}(z) = \prod_k \mathcal{N}(z_k; \bar{z}_k, a_k^2) \quad \text{(VAE case)}.$$

[*Method*] Variational Bayes Latent Estimation

Formulation with variational inference

- Suppose $p_{Y|Z}(y|z) = p_{Y|X}(y|x = D_\theta(z))$.
- Approximation of the measure posterior $p_{Z|Y}(z|y)$ with $q_{\bar{z},a}(z)$ with

$$\bar{z}, a \in \mathbb{R}^{C\times M\times N}, a > 0 \text{ and } q_{\bar{z},a}(z) = \prod_k \mathcal{U}(z_k; [\bar{z}_k - \frac{a_k}{2}, \bar{z}_k + \frac{a_k}{2}]) \quad \text{(CAE case)},$$

$$q_{\bar{z},a}(z) = \prod_k \mathcal{N}(z_k; \bar{z}_k, a_k^2) \quad \text{(VAE case)}.$$

- Minimize $KL(q_{\bar{z},a}(z)||p_{Z|Y}(z|y)) \Rightarrow$ derivation of the ELBO

$$\arg\max_{\bar{z},a} \mathcal{L}_{\bar{z},a} = \arg\max_{z,a} \mathbb{E}_{q_{\bar{z},a}(z)}\big[\log p_{Y|Z}(y|z) + \log p_\theta(z) - \log q_{\bar{z},a}(z)\big] \qquad (1)$$

[*Method*] Final algorithm

Reparameterization trick: $q_{\bar{z},a} = \bar{z} + a\epsilon$

$\rightarrow$ Stochastic Gradient Variational Bayes (SGVB) estimate can be derived[1]

Algorithm Variational Bayes Latent Estimation

Require: $\bar{z}_0 \in \mathbb{R}^{C\times M\times N}$, $a_0 \in \mathbb{R}^{C\times M\times N} = (1)_{i,j,l}$, $k = 0$, $\eta > 0$

while not *convergence* **do**

$z_1 \sim q_{\bar{z}_k,a_k}(z_1), \ldots, z_n \sim q_{\bar{z}_k,a_k}(z_n)$

$\begin{pmatrix} \bar{z}_{k+1} \\ a_{k+1} \end{pmatrix} = \begin{pmatrix} \bar{z}_k \\ a_k \end{pmatrix} - \eta \nabla_{\bar{z},a} \frac{1}{n} \sum_{i=1}^{n} \left[-\log p_{Y|Z}(y|z_i) - \log p_\theta(z_i) + \log q_{\bar{z},a}(z_i)) \right]$

$k = k + 1$

end while

return $(\bar{z}^*, a^*) = (\bar{z}_k, a_k)$

[1] D.P. Kingma, M.Welling, *Auto-Encoding Variational Bayes*, ICLR 2014

[*Method*] Final algorithm

Reparameterization trick: $q_{\bar{z},a} = \bar{z} + a\epsilon$

$\rightarrow$ Stochastic Gradient Variational Bayes (SGVB) estimate can be derived[1]

Algorithm Variational Bayes Latent Estimation

Require: $\bar{z}_0 \in \mathbb{R}^{C\times M\times N}$, $a_0 \in \mathbb{R}^{C\times M\times N} = (1)_{i,j,l}$, $k = 0$, $\eta > 0$
  while not *convergence* **do**
    $z_1 \sim q_{\bar{z}_k,a_k}(z_1), \dots, z_n \sim q_{\bar{z}_k,a_k}(z_n)$
    $\begin{pmatrix} \bar{z}_{k+1} \\ a_{k+1} \end{pmatrix} = \begin{pmatrix} \bar{z}_k \\ a_k \end{pmatrix} - \eta \nabla_{\bar{z},a} \frac{1}{n} \sum_{i=1}^{n} \Big[- \log p_{Y|Z}(y|z_i) - \log p_\theta(z_i) + \log q_{\bar{z},a}(z_i)) \Big]$
    $k = k + 1$
  end while
  return $(\bar{z}^*, a^*) = (\bar{z}_k, a_k)$

Final MMSE estimation:

$$x^*_{MMSE-z} = D_\theta(\bar{z}^*) \quad \text{or} \quad x^*_{MMSE-x} = \frac{1}{L}\sum_{i=1}^{L} D_\theta(z_i) \text{ with } z_i \sim q_{\bar{z}^*,a^*}(z_i).$$

[1] D.P. Kingma, M.Welling, *Auto-Encoding Variational Bayes*, ICLR 2014

[*SatImg*] Data setup

- Pelican (10cm) and PCRS (5cm) airborne images
- Downsampled ⇒ "ideal" satellite images.
- Pléiades operating point (50cm)

Datasets

- Train images : from Pelican and/or PCRS images downsampled to a given resolution.
- Test images : Subpart of 14 Pelican images, simulated with CNES CSI (Chaîne de Simulation Image).

Figure: Example of a satellite image ©CNES 2024

[*SatImg*] Problem statement

Forward model :

$$y = h * x + n \text{ with } n \sim \mathcal{N}(0, \sigma_x^2 = a + b(h * x)),$$
$$\tilde{y} = Q(y) \text{ (compression-decompression)}$$

- h : PSF (+ downsampling operator if super-resolution)

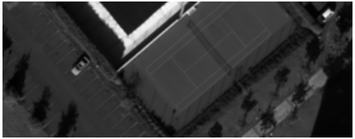


Figure: Target (25cm) ©CNES 2024

Figure: Target (50cm) ©CNES 2024

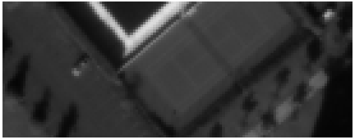


Figure: Measure (50cm) ©CNES 2024

[*SatImg*] Method recap

Step 1 : Training of an autoencoder on ideal images

- At 50cm (blur+noise)
- Or at 25cm when super resolution (blur+noise+SISR)

Etape 2 : Image restoration

- VBLE minimization
- $||AG(z) - y||^2$ becomes $\frac{1}{2}||y - h * G(z)||^2_{\Sigma_x^{-1}} + \frac{1}{2}\log|\Sigma_x|$

[SatImg] Some visual results

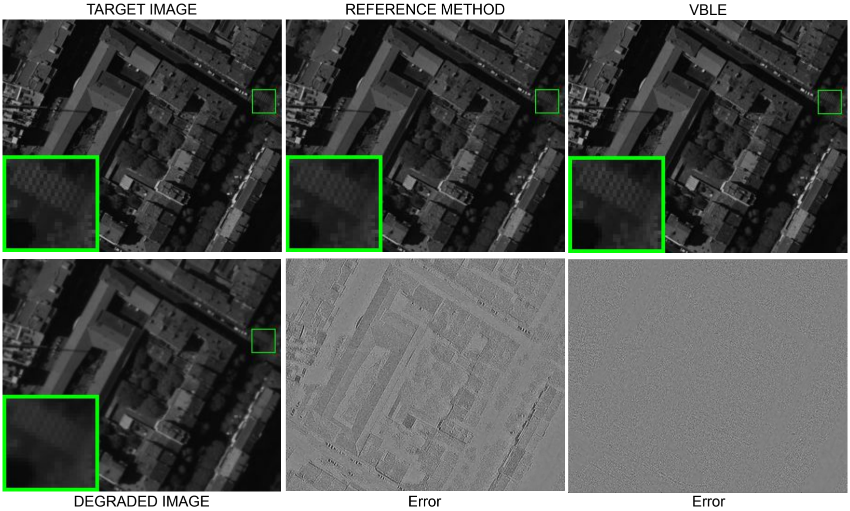


Figure: ©CNES 2024

[SatImg] Some visual results - with super resolution

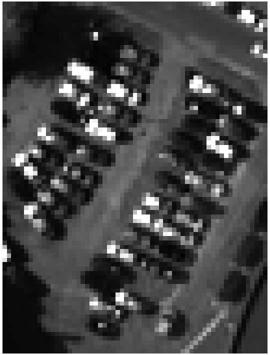


Figure: IR at 50cm

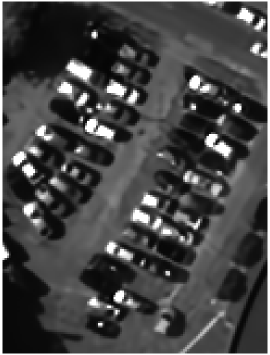


Figure: IR at 25cm

Figure: Target

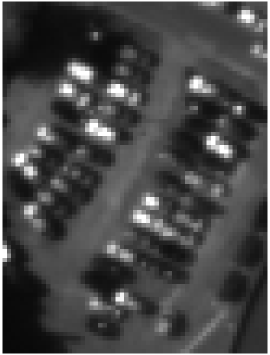


Figure: y

[SatImg] VBLE - Sampling

Figure: VBLE MMSE-x

Figure: VBLE sample

Figure: Target

Figure: y

[SatImg] VBLE - Sampling

Figure: VBLE MMSE-x

Figure: VBLE sample

Figure: Target

Figure: y

[SatImg] VBLE - Sampling

Figure: VBLE MMSE-x

Figure: VBLE std

Figure: Target

Figure: y

[SatImg] Robustness - resolution

- Training of two networks (25cm and 50cm)
- Restoration at 50cm (blur+noise) and/or 25cm (blur+noise+SISR)
- DPIR[1] : a state-of-the-art plug-and-play method

	VBLE 50cm	VBLE 25cm	DPIR 50cm	DPIR 25cm
PSNR ↑	47.36	47.27	48.05	48.11
SSIM ↑	0.9941	0.9939	0.9952	0.9952
LPIPS ↓	0.0235	0.0243	0.0169	0.0168

Table: Image restoration at 50cm (blur+noise) with different networks

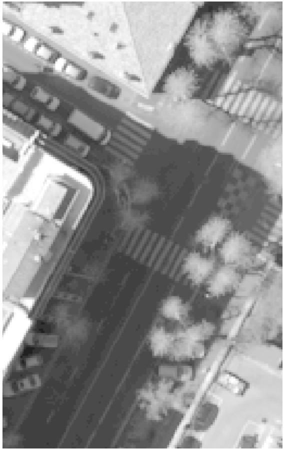
[1] Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R., *Plug-and-play image restoration with deep denoiser prior*, IEEE TPAMI 2021

[SatImg] Robustness - resolution

Restoration with super resolution (at 25cm)

Figure: DPIR 25cm

Figure: DPIR 50cm



Target

[*SatImg*] Robustness - landscape

- Training of two networks (PCRS or Pélican datasets)
- Image restoration of Pélican test dataset.

	DPIR		**VBLE**	
	Pélican	PCRS	Pélican	PCRS
PSNR	48.36	48.27	48.21	47.96
SSIM	0.9959	0.9959	0.9956	0.9954
LPIPS	0.0151	0.0162	0.0211	0.0216

Table: Restoration at 50cm (blur + noise) for DPIR and VBLE on Pélican, with networks trained on Pélican or PCRS.

- Visually, no significant differences.

Conclusion

VBLE

- Performs variational inference in the latent space of VAEs to approximate the posterior distribution $p(z|y)$
- Efficient for posterior sampling
- Works well on satellite images

Perspectives

- Uncertainty evaluation and calibration
- Application to other inverse problems

[*Appendice*] Bora's method as latent MAP

Optimization problem

$$\hat{z} = \underset{z}{\operatorname{argmin}} ||AG(z) - y||^2 + \lambda ||z||^2$$

Corresponds to a latent Maximum A Posteriori (MAP):

$$\hat{z} = \underset{z}{\operatorname{argmin}} - \log p_{Y|Z}(y|z) - \log p_Z(z) = \underset{z}{\operatorname{argmax}}\, p_{Z|Y}(z|y)$$

With

- a Gaussian prior ($p(z) = \mathcal{N}(0, I)$, $\lambda = \frac{1}{2}$)
- a Gaussian noise model ($p_{Y|X}(y|x) = \mathcal{N}(y; Ax, \sigma^2 I)$)
- a deterministic generator ($p_{Y|Z}(y|z) = p_{Y|X}(y|x = G(z))$)

[SatImg] Intuition

Representation error : the target x not necessarily in $\{G(z) | z \in Z\}$

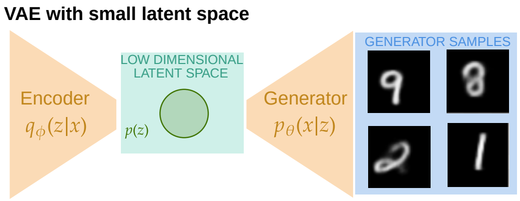


Figure: Top: Classical VAE. Bottom: Used autoencoder.

[SatImg] Intuition

Representation error : the target x not necessarily in $\{G(z)|z \in Z\}$

Use of an autoencoder with...

- a wider latent space,

$\Rightarrow$ a wider range for the decoder,

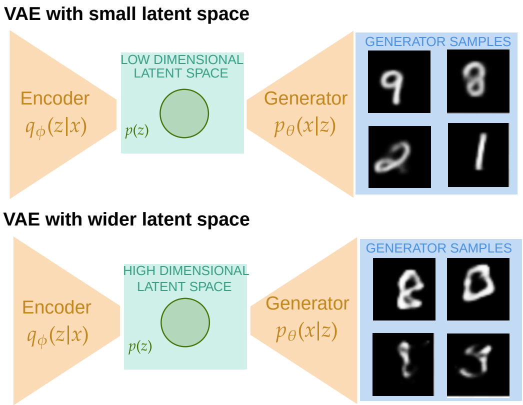


Figure: Top: Classical VAE. Bottom: Used autoencoder.

[SatImg] Intuition

Representation error : the target x not necessarily in $\{G(z)|z \in Z\}$

Use of an autoencoder with...

- a wider latent space,

$\Rightarrow$ a wider range for the decoder,

- less structure in the latent space.

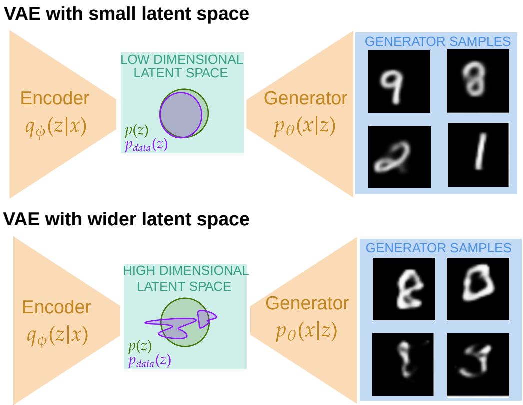


Figure: Top: Classical VAE. Bottom: Used autoencoder.

[*SatImg*] Intuition

Representation error : the target x not necessarily in $\{G(z)|z \in Z\}$

Use of an autoencoder with...

- a wider latent space,

$\Rightarrow$ a wider range for the decoder,

- less structure in the latent space.

Original restoration process:
$\arg\min_z ||AG(z) - y||^2 + \lambda||z||^2$

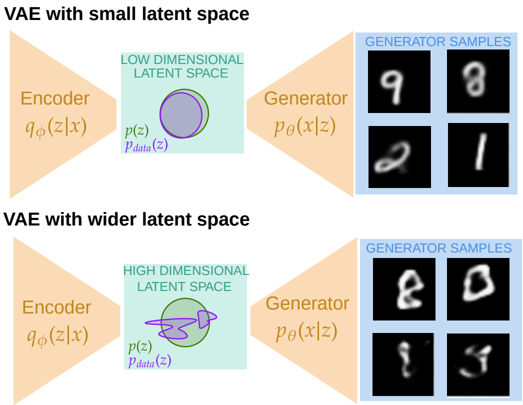


Figure: Top: Classical VAE. Bottom: Used autoencoder.

[*SatImg*] Intuition

Representation error : the target x not necessarily in $\{G(z)|z \in Z\}$

Use of an autoencoder with...

- a wider latent space,

$\Rightarrow$ a wider range for the decoder,

- less structure in the latent space.

Modified restoration process:
$\arg\min_z ||AG(z) - y||^2 + \lambda R(z)$

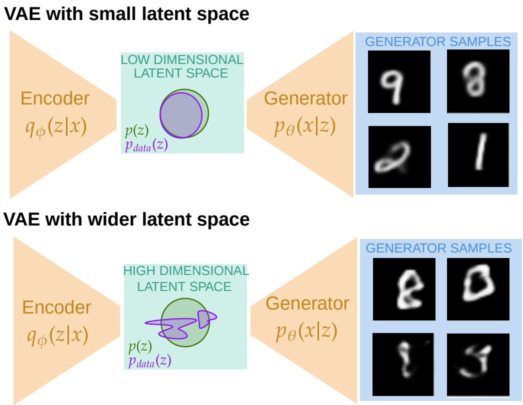


Figure: Top: Classical VAE. Bottom: Used autoencoder.

[SatImg] Use of a compressive autoencoder (CAE)

Motivation

- Lightweight neural architecture
- Can be seen as VAEs

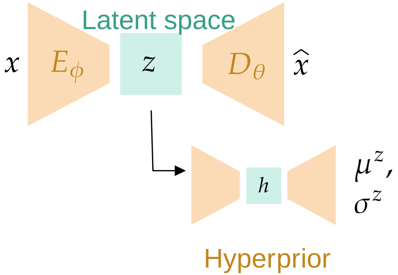


Figure: Compressive autoencoder with hyperprior[1]

[1] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, *Variational image compression with a scale hyperprior*, ICLR 2018

[SatImg] Use of a compressive autoencoder (CAE)

Motivation

- Lightweight neural architecture
- Can be seen as VAEs
- The hyperprior = adaptive prior on the latent distribution $\rightarrow p(z) \propto \mathcal{N}(z; \mu^z, \sigma^z)$.

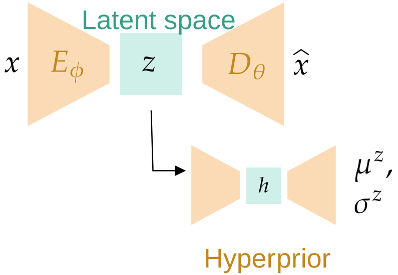


Figure: Compressive autoencoder with hyperprior[1]

[1] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, *Variational image compression with a scale hyperprior*, ICLR 2018

[*SatImg*] Use of a compressive autoencoder (CAE)

Motivation

- Lightweight neural architecture
- Can be seen as VAEs
- The hyperprior = adaptive prior on the latent distribution $\rightarrow p(z) \propto \mathcal{N}(z; \mu^z, \sigma^z)$.

Latent MAP estimation for a CAE:

$$\hat{z} = \underset{z}{\operatorname{argmin}} ||AD_\theta(z) - y||^2 + \lambda R(z)$$

with $\lambda R(z) \propto -\log p(z)$.

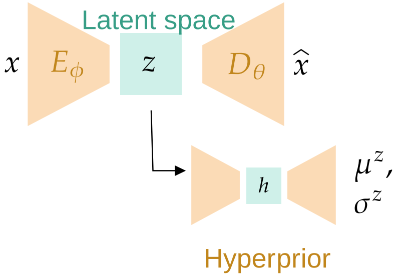


Figure: Compressive autoencoder with hyperprior[1]

[1] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, *Variational image compression with a scale hyperprior*, ICLR 2018

[Appendice] Use of a compressive autoencoder (CAE) - Details

CAEs can generally be expressed as VAEs.

- Hyperprior $\simeq$ 2 latent variable VAE with

$$p_\theta(z, h) = p_\theta(z|h)p_\theta(h)$$

and $$p_\theta(z|h) = \prod_k \left[\mathcal{N}(\mu^z, \sigma^z) * \mathcal{U}[-\frac{1}{2}, \frac{1}{2}])\right](z_k)$$

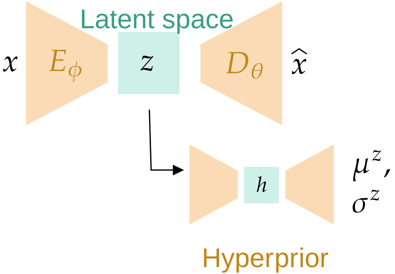


Figure: Compressive autoencoder with hyperprior[1]

[1] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, *Variational image compression with a scale hyperprior*, ICLR 2018

[Appendice] Use of a compressive autoencoder (CAE) - Details

CAEs can generally be expressed as VAEs.

- Hyperprior $\simeq$ 2 latent variable VAE with

$$p_\theta(z,h) = p_\theta(z|h)p_\theta(h)$$

$$\text{and } p_\theta(z|h) = \prod_k \left[\mathcal{N}(\mu^z,\sigma^z) * \mathcal{U}[-\frac{1}{2},\frac{1}{2}])\right](z_k)$$

- Uniform encoder posterior distribution

$$q_\phi(z,h|x) = q_\phi(z|x,h)q_\phi(h|x)$$

$$\text{with } q_\phi(z|x,h) = \prod_k \mathcal{U}([z_k - \frac{1}{2}, z_k + \frac{1}{2}])$$

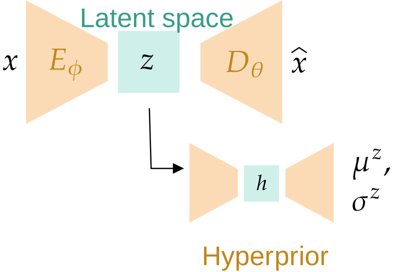


Figure: Compressive autoencoder with hyperprior[1]

[1] J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, *Variational image compression with a scale hyperprior*, ICLR 2018

[Appendice] VBLE results - FFHQ results

Chosen baselines

- DiffPIR[1] (with diffusion models)
- Latent MAP with a single latent variable VAE (MAPz wVAE-V1 and V2)

FFHQ	Deblur (motion)			SISR $\times 4$			Inpainting (random)		
Method	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
VBLE wCAE	**32.08**	0.2010	**0.8751**	**31.26**	0.2048	**0.8699**	**36.98**	0.0795	**0.9623**
MAP-z wCAE	32.01	0.2441	0.8717	31.08	0.2205	0.8681	36.91	0.0715	0.9618
MAP-z wVAE V1	27.95	0.4331	0.6582	30.24	0.2201	0.8499	35.21	0.0775	0.9443
MAP-z wVAE V2	29.44	0.3088	0.8112	29.53	0.2765	0.8226	31.01	0.2660	0.8513
DiffPIR	31.98	**0.1793**	0.8677	30.71	**0.2042**	0.8492	36.08	**0.0653**	0.9470

Table: FFHQ results on: motion deblurring ($= 7.65/255$), SISR $\times 4$, inpainting with random masks.

[1] Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., Van Gool, L., *Denoising Diffusion Models for Plug-and-Play Image Restoration*, CVPR 2023

[Appendice] VBLE results - FFHQ visual results

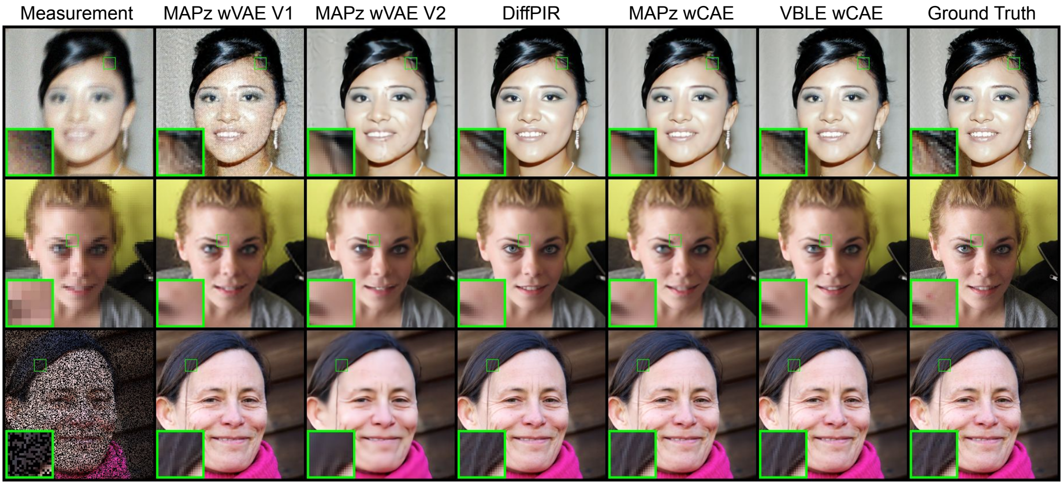


Figure: Qualitative image restoration results on FFHQ. From top to bottom: Gaussian deblurring ($\sigma_{blur} = 1, \sigma = 7.65/255$), SISR $\times 2$, inpainting.

[Appendice] VBLE results - BSD visual results

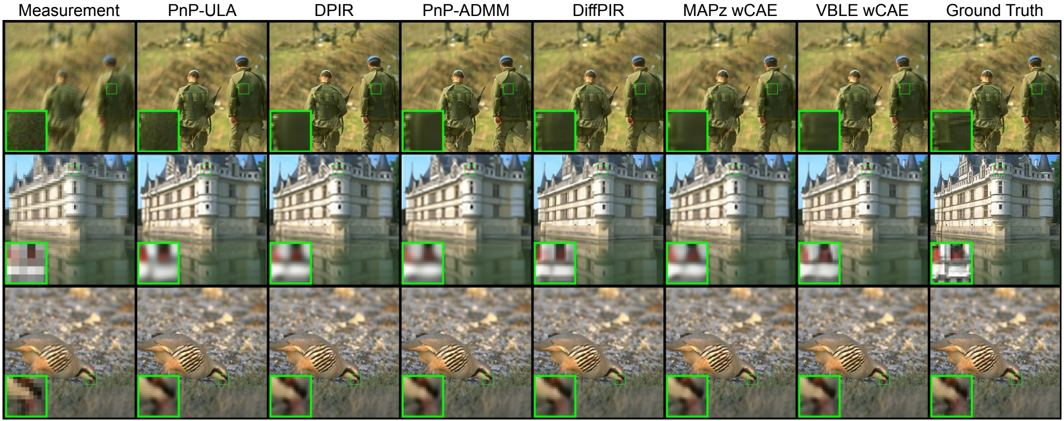


Figure: Qualitative image restoration results on BSD. From top to bottom: Motion deblurring ($\sigma = 7.65/255$), SISR $\times 4$, SISR $\times 2$.

[Appendice] VBLE results - Posterior distribution quality assessment

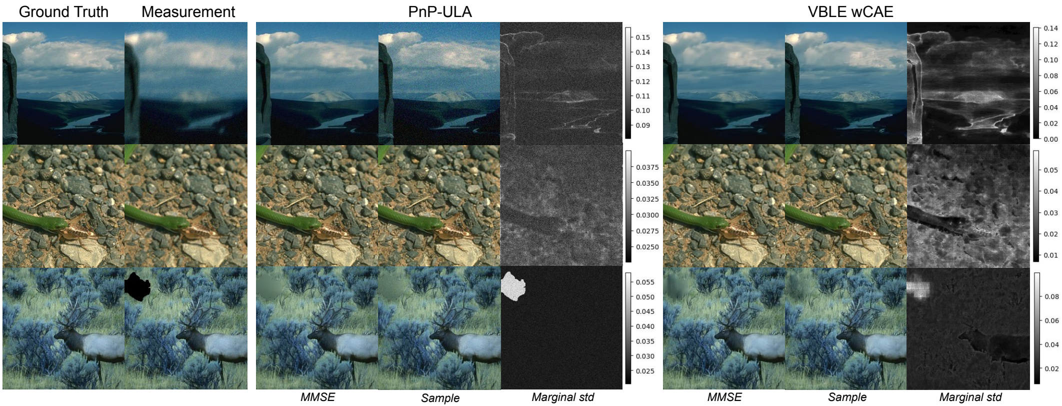


Figure: Visual comparison between VBLE and PnP-ULA on several inverse problems. From top to bottom: Motion deblurring ($\sigma = 7.65/255$), SISR $\times 2$, block inpainting ($\sigma = 2.55/255$).

[Appendice] VBLE results - Posterior distribution quality assessment (2)

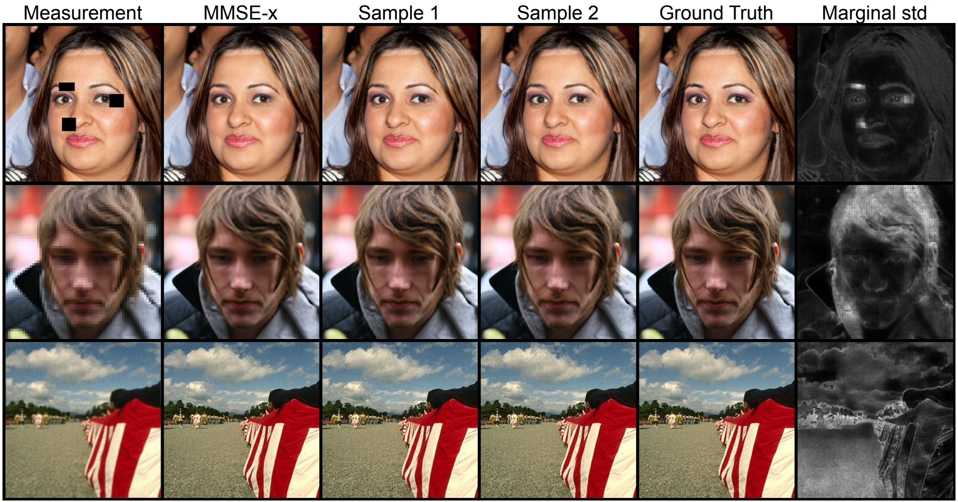


Figure: Example of VBLE posterior sampling ability. The MMSE-x, which is the estimate used in all quantitative comparison, corresponds to the average over samples in the image space.

[*Appendice*] Some results on satellite images

	Without SISR (50cm)		With SISR (25cm)	
	DPIR	VBLE	DPIR	VBLE
PSNR $\uparrow$	48.36	48.21	38.25	38.13
SSIM $\uparrow$	0.9959	0.9956	0.9581	0.9578
LPIPS $\downarrow$	0.0151	0.0211	0.1550	0.1605
Time (1img)	3s	31s	1min02s	27min48s

Table: DPIR and VBLE results on a test set of 14 satellite images simulated with Pelican images.

- Without specific proximal adaptation, time $\sim$ 30min for DPIR at 25cm.

[*Appendice*] Coverage

A l'aide de graphe de coverage

- On construit une région de confiance C_α pour l'image (ou pour chaque pixel) de niveau α à partir des échantillons de VBLE

$$C_\alpha = \{x \in X | ||x - \hat{x}||^2 < q_\alpha\}$$

avec q_α l'α-quantile de $(||x_1 - \hat{x}||^2, ..., ||x_n - \hat{x}||^2)$

$\Rightarrow$ $P(x^* \in C_\alpha) \approx \alpha$.

- On estime $\hat{\alpha} = \frac{1}{N} \sum_k (x_k^* \in C_\alpha)$
- Graphe de coverage : $\hat{\alpha}$ en fonction de α.

[*Appendice*] VBLE - Validité des incertitudes

Erreurs de VBLE

- Optimisation $\rightarrow$ supposée faible
- Problème inverse $\rightarrow$ modélisée par VBLE
- Représentation $\rightarrow$ difficilement modélisable

Modélisation par une variance additionnelle dans l'espace image. Génération d'une solution :

- $z = \bar{z} + a * u, \quad u \sim \mathcal{U}(-0.5, 0.5)$
- $x = D(z) + \sigma(z)\epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$