Structured Generative Models as Priors for Inverse Problems

Neill Campbell

Working with Era Dorta, Margaret Duff, Ivor Simpson, Sara Vicente, Lourdes Agapito, and Matthias Ehrhardt

Department of Computer Science, University of Bath

THE ROYAL SOCIETY · CDE Centre for Digital Entertainment · CAMERA Centre for the Analysis of Motion, Entertainment Research and Applications · UNIVERSITY OF BATH

Contents i

Here be monsters...

Storytime...

“breaking the ubiquitous ML assumption in image and vision computing that errors and uncertainties at neighbouring pixels are independent, despite their demonstrable spatial structure”

Is unsupervised learning a thing?

Figure 2: Stable Diffusion: *"The manifold of cats."*

Figure 2: Stable Diffusion: *"The manifold of cats."*

- "Find me some $p(\mathbf{z})$ and $f(\mathbf{z})$ such that $\mathbf{x} \sim f(\mathbf{z})$ when $\mathbf{z} \sim p(\mathbf{z})$.."

Figure 2: Stable Diffusion: *"The manifold of cats."*

- "Find me some $p(\mathbf{z})$ and $f(\mathbf{z})$ such that $\mathbf{x} \sim f(\mathbf{z})$ when $\mathbf{z} \sim p(\mathbf{z})$.."
- This has trivial solutions

Figure 2: Stable Diffusion: *"The manifold of cats."*

- "Find me some $p(\mathbf{z})$ and $f(\mathbf{z})$ such that $\mathbf{x} \sim f(\mathbf{z})$ when $\mathbf{z} \sim p(\mathbf{z})$.."
- This has trivial solutions
  - Need constraints

Figure 2: Stable Diffusion: *"The manifold of cats."*

- "Find me some $p(\mathbf{z})$ and $f(\mathbf{z})$ such that $\mathbf{x} \sim f(\mathbf{z})$ when $\mathbf{z} \sim p(\mathbf{z})$.."
- This has trivial solutions
  - Need constraints
  - Utility $\leftrightarrow$ use-case

Figure 2: Stable Diffusion: *"The manifold of cats."*

- "Find me some $p(\mathbf{z})$ and $f(\mathbf{z})$ such that $\mathbf{x} \sim f(\mathbf{z})$ when $\mathbf{z} \sim p(\mathbf{z})$.."
- This has trivial solutions
  - Need constraints
  - Utility $\leftrightarrow$ use-case
- Generative models as **priors**

Inverse problem setup

- Inverse problem $\mathbf{y} = A\,\mathbf{x} + \varepsilon$ for some forward model $A : \mathcal{X} \to \mathcal{Y}$ and noise ε

Inverse problem setup

- Inverse problem $\mathbf{y} = A\,\mathbf{x} + \varepsilon$ for some forward model $A : \mathcal{X} \to \mathcal{Y}$ and noise ε
- *Variational* regularisation framework (for some similarity $D(\cdot, \cdot)$)

$$\mathbf{x}^* \in \arg\min_{\mathbf{x} \in \mathcal{X}} D(\mathbf{y}, A\,\mathbf{x}) + \lambda\, R(\mathbf{x})$$

Inverse problem setup

- Inverse problem $\mathbf{y} = A\,\mathbf{x} + \varepsilon$ for some forward model $A : \mathcal{X} \to \mathcal{Y}$ and noise ε
- *Variational* regularisation framework (for some similarity $D(\cdot,\cdot)$)

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} D(\mathbf{y}, A\,\mathbf{x}) + \lambda\, R(\mathbf{x})$$

- Regulariser from an *explicit* prior distribution, $R(\mathbf{x}) := \log p(\mathbf{x} \,|\, \boldsymbol{\theta})$

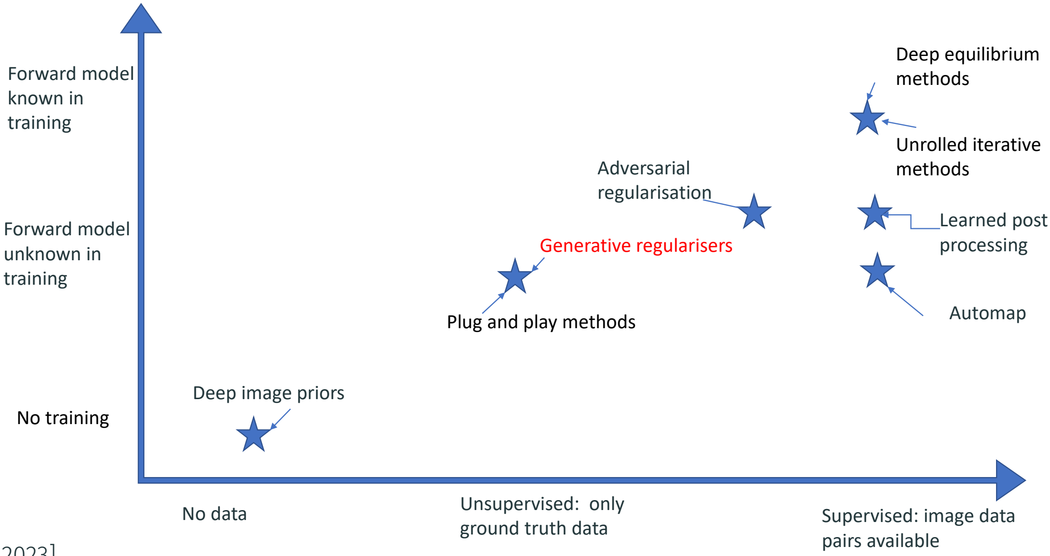
Inverse problem setup

- Inverse problem $\mathbf{y} = A\,\mathbf{x} + \varepsilon$ for some forward model $A : \mathcal{X} \to \mathcal{Y}$ and noise ε
- *Variational* regularisation framework (for some similarity $D(\cdot,\cdot)$)

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} D(\mathbf{y}, A\,\mathbf{x}) + \lambda\, R(\mathbf{x})$$

- Regulariser from an *explicit* prior distribution, $R(\mathbf{x}) := \log p(\mathbf{x}\,|\,\boldsymbol{\theta})$
- $\mathbf{x}^*$ considered a MAP estimate if $D(\mathbf{y}, A\,\mathbf{x}) := \log p(\mathbf{y}\,|\,f(A\,\mathbf{x}), \ldots)$

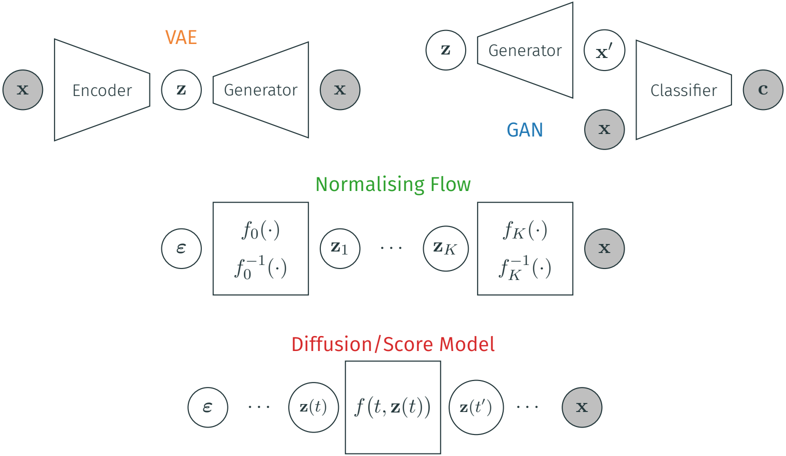
Deep learning approaches for inverse problems



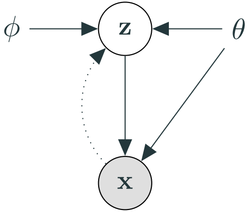
[Duff 2023]

Generative models

Generative model zoo



Unreasonable expectations of generative models?



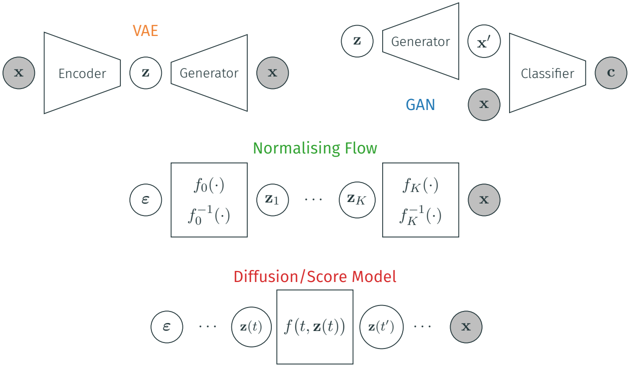
e.g. VAE with:

$$\mathbf{z} \in \mathbb{R}^M,$$
$$\mathbf{x} \in [0,1]^{3\times N\times N}$$

Figure 3: How many degrees of freedom are there in the image?

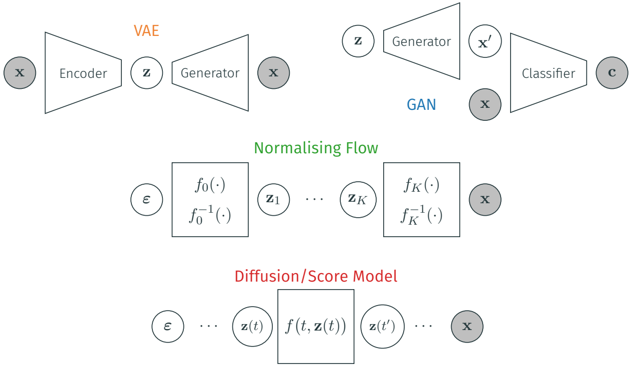
Properties we would like

- Span the data space



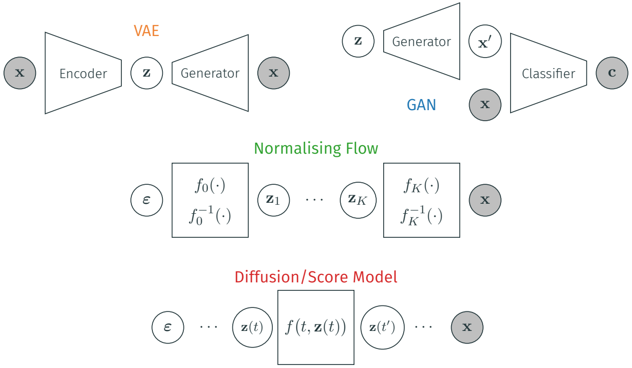
Properties we would like

- Span the data space
- Representative samples



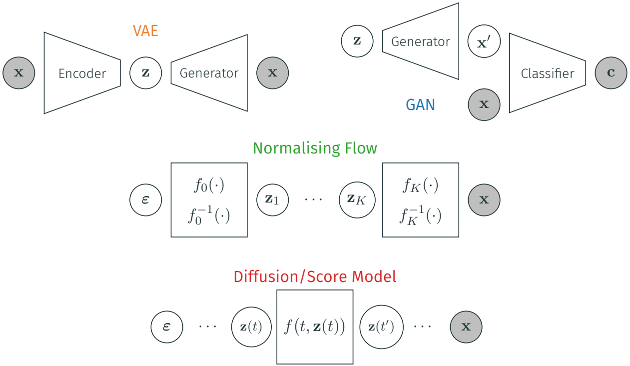
Properties we would like

- Span the data space
- Representative samples
- Conditions on mapping (e.g. “smooth”)



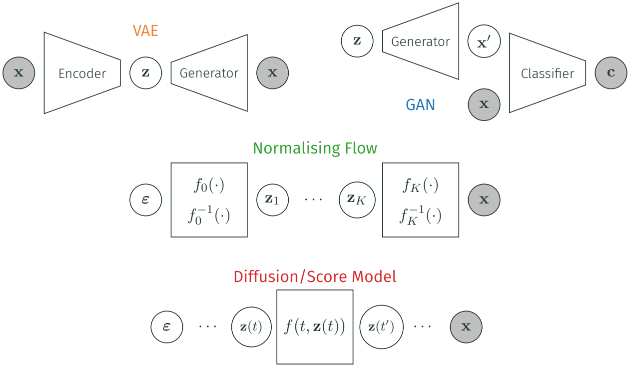
Properties we would like

- Span the data space
- Representative samples
- Conditions on mapping (e.g. “smooth”)
- Evaluate densities (e.g. take likelihood)



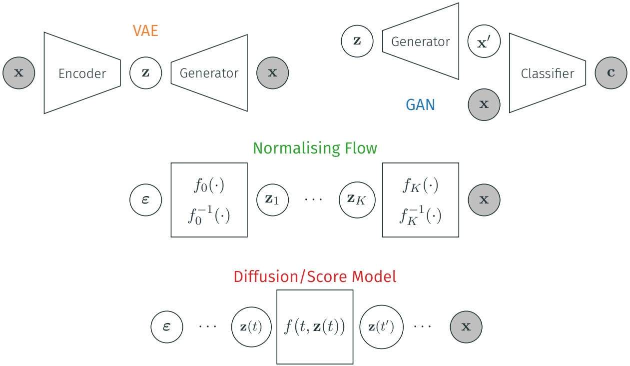
Properties we would like

- Span the data space
- Representative samples
- Conditions on mapping (e.g. “smooth”)
- Evaluate densities (e.g. take likelihood)
- Uncertainty (e.g. account for failure to model)



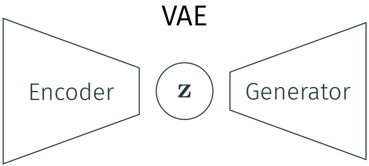
Properties we would like

- Span the data space
- Representative samples
- Conditions on mapping (e.g. “smooth”)
- Evaluate densities (e.g. take likelihood)
- Uncertainty (e.g. account for failure to model)
- Introspection



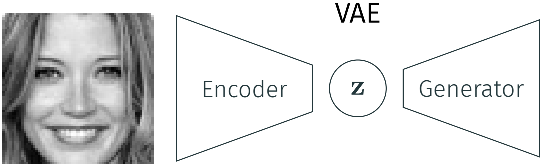
Structured Uncertainty Prediction Networks (SUPN)

“VAEs produce overly smooth output”



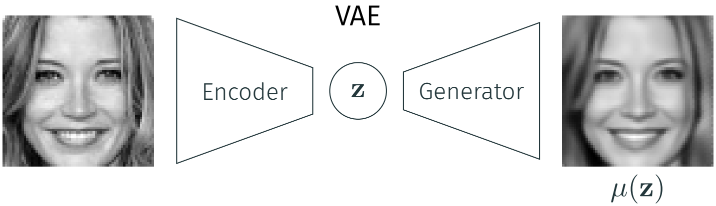
[Dorta et al. 2018]

“VAEs produce overly smooth output”



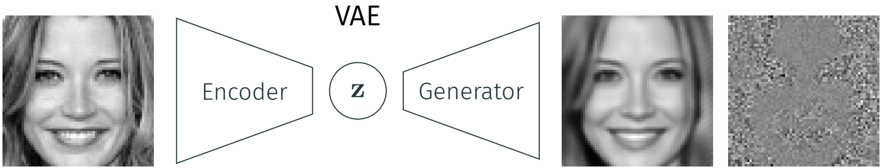
[Dorta et al. 2018]

“VAEs produce overly smooth output”



[Dorta et al. 2018]

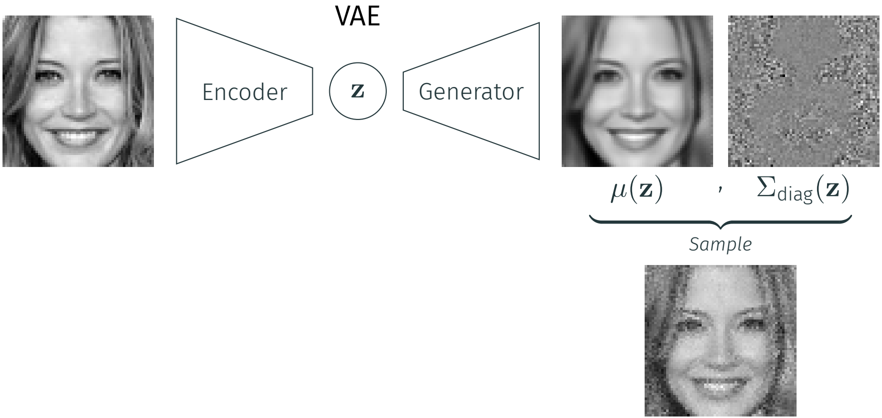
“VAEs produce overly smooth output”



$\mu(\mathbf{z})$, $\Sigma_{\text{diag}}(\mathbf{z})$

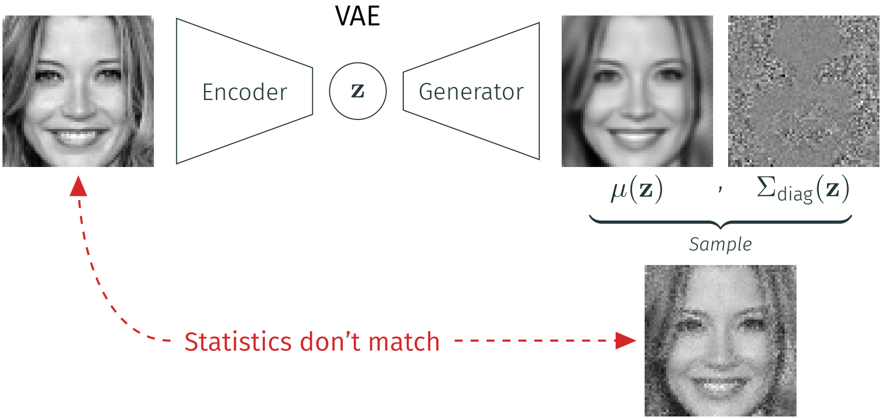
[Dorta et al. 2018]

“VAEs produce overly smooth output”



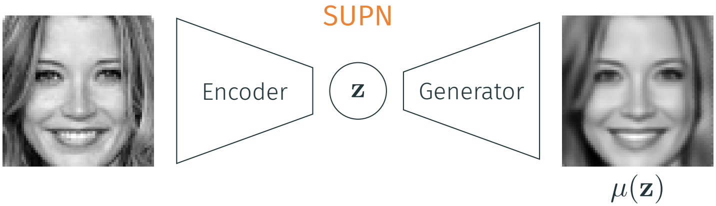
[Dorta et al. 2018]

"VAEs produce overly smooth output"



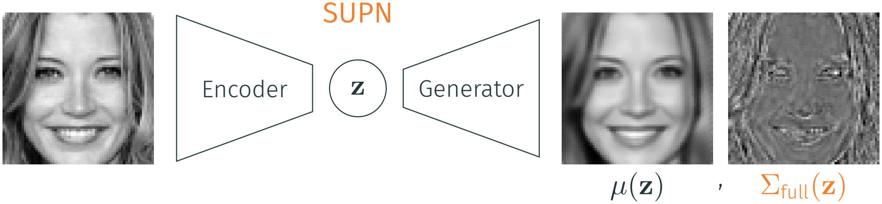
[Dorta et al. 2018]

“VAEs produce overly smooth output”



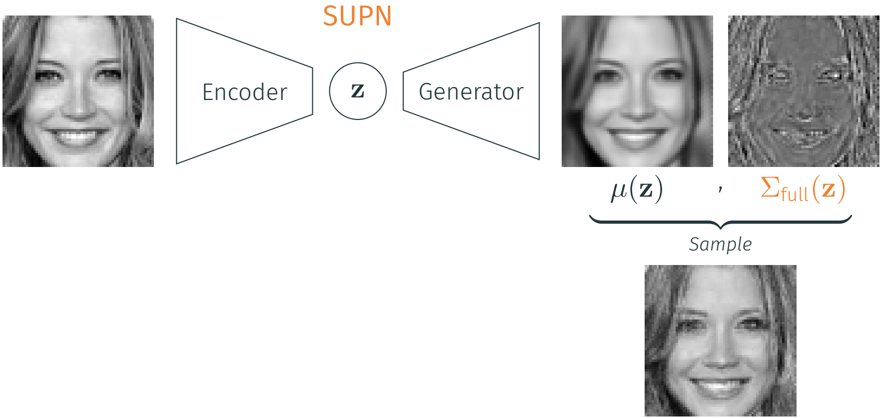
[Dorta et al. 2018]

“VAEs produce overly smooth output”

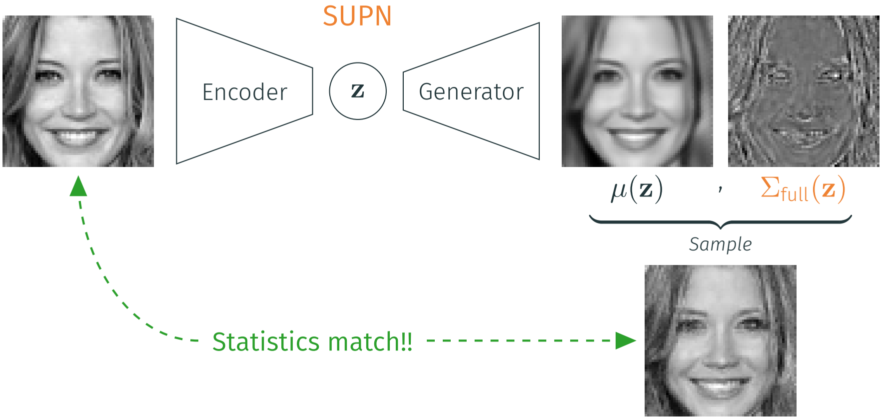


[Dorta et al. 2018]

“VAEs produce overly smooth output”

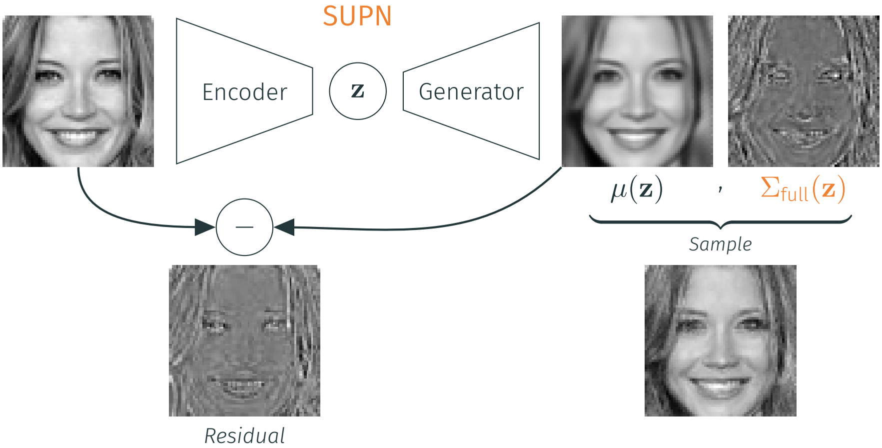


[Dorta et al. 2018]

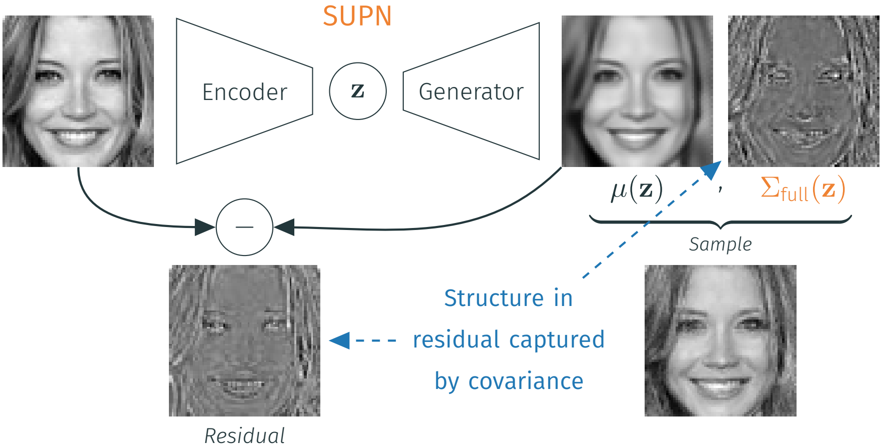


[Dorta et al. 2018]

“VAEs produce overly smooth output”



[Dorta et al. 2018]



[Dorta et al. 2018]

Problem! Dense covariance $\mathcal{O}(N^2)$...

- **Problem**: $\Sigma_{\text{full}}(\mathbf{z})$ is quadratic in the number of pixels

Problem! Dense covariance $\mathcal{O}(N^2)$...

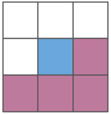
- **Problem**: $\Sigma_{\text{full}}(\mathbf{z})$ is quadratic in the number of pixels
- **Solution**: Sparse parameterisation of the Cholesky factor of the precision

$$\Sigma(\mathbf{z}) := [\Lambda(\mathbf{z})]^{-1} := [L_\Lambda(\mathbf{z})\, L_\Lambda^\top(\mathbf{z})]^{-1}$$

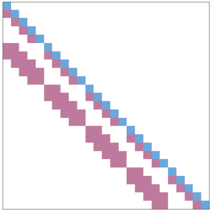
Problem! Dense covariance $\mathcal{O}(N^2)$...

- **Problem**: $\Sigma_{\text{full}}(\mathbf{z})$ is quadratic in the number of pixels
- **Solution**: Sparse parameterisation of the Cholesky factor of the precision

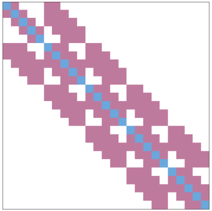
$$\Sigma(\mathbf{z}) := [\Lambda(\mathbf{z})]^{-1} := [L_\Lambda(\mathbf{z})\, L_\Lambda^\top(\mathbf{z})]^{-1}$$



Neighbourhood in image domain



Sparsity in the precision Cholesky matrix L_Λ



Sparsity in the precision matrix $\Lambda(\mathbf{z}) := \Sigma^{-1}(\mathbf{z})$

Efficient implementation

- Sparse parameterisation of the Cholesky factor of the precision

$$\Sigma(\mathbf{z}) := [\Lambda(\mathbf{z})]^{-1} := [L_\Lambda(\mathbf{z})\, L_\Lambda^\top(\mathbf{z})]^{-1}$$

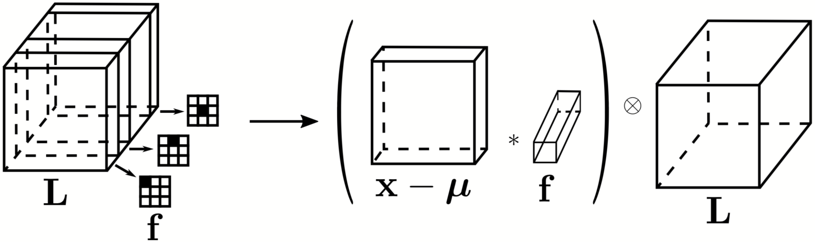


Figure 4: Implementation through convolutional structure: matrix-vector product in $\mathcal{O}(N)$

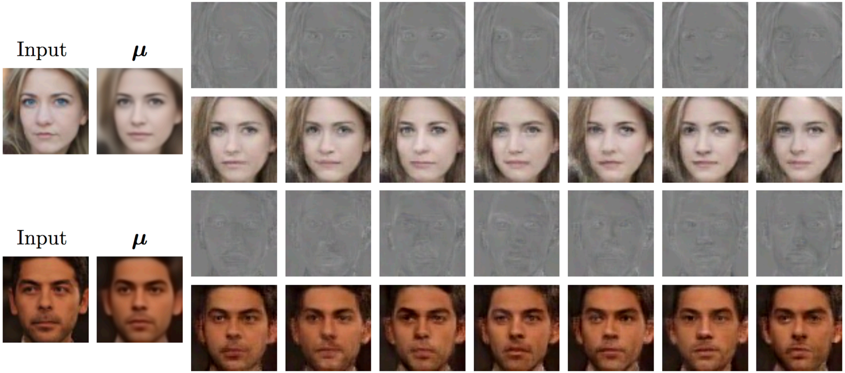


Figure 5: Variation in samples from the model on test data

Introspection of the captured covariance structure

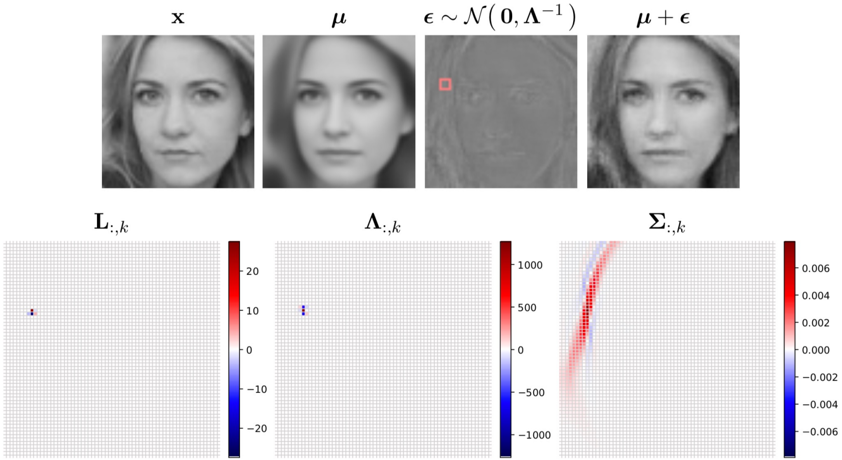


Figure 6: Visualisation of the learned correlations

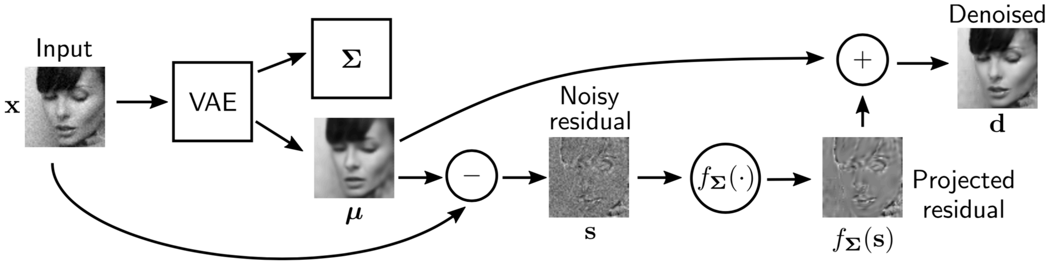
Links to established concepts...

- Links to Conditional Random Field (CRF) models
  - a Gaussian CRF - e.g. “Regression Tree Fields” [Jancsary et al. 2012]
- Links to adaptive local regularisation models
  - e.g. locally adaptive TV or Laplacian based methods
- Links to Wavelet approaches
  - considering hierarchical extensions or combining fixed basis functions

Links to established concepts…

- Links to Conditional Random Field (CRF) models
  - a Gaussian CRF - e.g. “Regression Tree Fields” [Jancsary et al. 2012]
- Links to adaptive local regularisation models
  - e.g. locally adaptive TV or Laplacian based methods
- Links to Wavelet approaches
  - considering hierarchical extensions or combining fixed basis functions

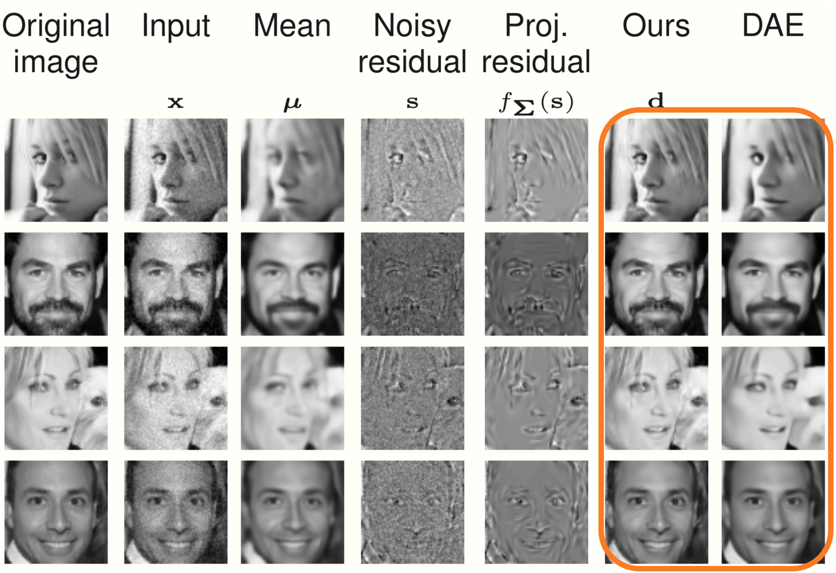
- Things to be careful about
  - priors on sparse precision (consider Cholesky structure)
  - need to bound terms
  - *lots to say about these things…*



Model	MSE	PSNR	SSIM
DAE	0.005 ± 0.003	28.89 ± 1.69	0.90 ± 0.03
SUPN	$\mathbf{0.003 \pm 0.001}$	$\mathbf{31.38 \pm 0.92}$	$\mathbf{0.92 \pm 0.02}$

Figure 7: Denoising example using SUPN (vs a denoising autoencoder). The SUPN model has only been trained as in a generative manner (i.e. as a prior).

Testing with denoising...



SUPN as a prior for inverse problems

SUPN as a prior for inverse problems

- Consider a hierarchical model for the inverse problem

$$p(\mathbf{x}, \mathbf{z} \,|\, \mathbf{y}) \propto p(\mathbf{y} \,|\, \mathbf{x})\, p_{\mathcal{G}}(\mathbf{x} \,|\, \mathbf{z})\, p_{\mathcal{Z}}(\mathbf{z})$$

- We will take a MAP estimate for $\mathbf{z}$ rather than marginalising `:-(`

SUPN as a prior for inverse problems

- Consider a hierarchical model for the inverse problem

$$p(\mathbf{x}, \mathbf{z} \,|\, \mathbf{y}) \propto p(\mathbf{y} \,|\, \mathbf{x})\, p_{\mathcal{G}}(\mathbf{x} \,|\, \mathbf{z})\, p_{\mathcal{Z}}(\mathbf{z})$$

- We will take a MAP estimate for $\mathbf{z}$ rather than marginalising `:-(`
- From before (with a Gaussian observation likelihood)

$$D(\mathbf{y}, A\,\mathbf{x}) := \frac{1}{2\sigma^2}\|A\,\mathbf{x} - \mathbf{y}\|_2^2$$

$$R(\mathbf{x}) := \min_{\mathbf{z} \in \mathcal{Z}} \; \log|\Sigma_\theta(\mathbf{z})| + \frac{1}{2}\|\mathbf{x} - \mu_\theta(\mathbf{z})\|^2_{\Sigma_\theta(\mathbf{z})} + \frac{1}{2}\|\mathbf{z}\|_2^2$$

- Where the *Generator* provides $\mathcal{N}(\mathbf{x} \,|\, \mu_\theta(\mathbf{z}), \Sigma_\theta(\mathbf{z}))$ via a network $[\mu, L_\Lambda] = f(\mathbf{z}; \theta)$ and $\|\mathbf{a}\|^2_\Sigma := \mathbf{a}^\top \Sigma^{-1}\,\mathbf{a}$ denotes a Gaussian weighted norm

- Consider a hierarchical model for the inverse problem

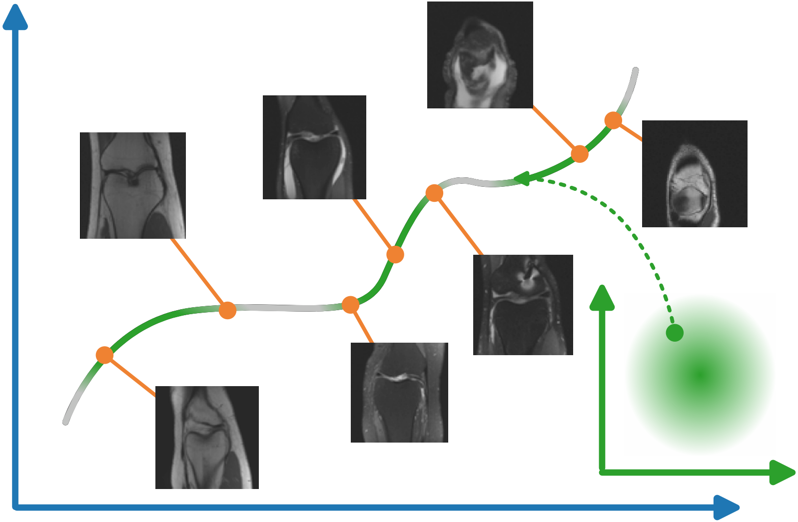
$$p(\mathbf{x}, \mathbf{z} \,|\, \mathbf{y}) \propto p(\mathbf{y} \,|\, \mathbf{x})\, p_{\mathcal{G}}(\mathbf{x} \,|\, \mathbf{z})\, p_{\mathcal{Z}}(\mathbf{z})$$

- We will take a MAP estimate for $\mathbf{z}$ rather than marginalising `:-(`
- From before (with a Gaussian observation likelihood)

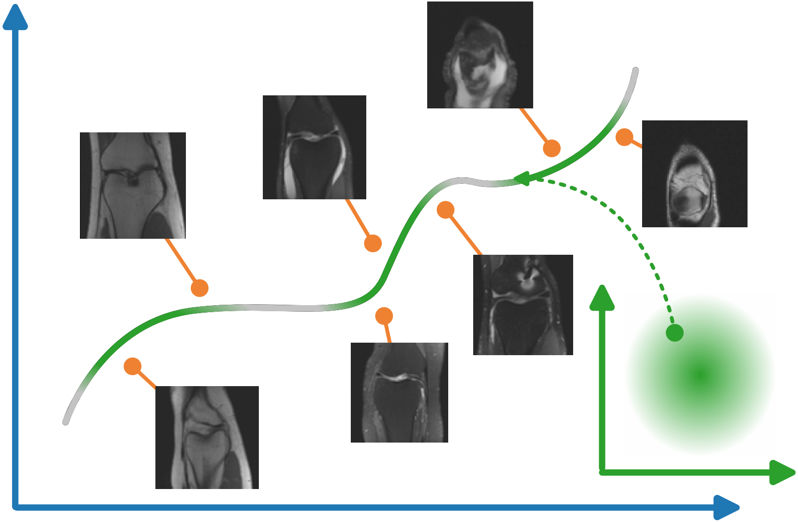
$$D(\mathbf{y}, A\,\mathbf{x}) := \frac{1}{2\sigma^2} \|A\,\mathbf{x} - \mathbf{y}\|_2^2$$

$$R(\mathbf{x}) := \min_{\mathbf{z} \in \mathcal{Z}} \; \log|\Sigma_\theta(\mathbf{z})| + \frac{1}{2}\|\mathbf{x} - \mu_\theta(\mathbf{z})\|^2_{\Sigma_\theta(\mathbf{z})} + \frac{1}{2}\|\mathbf{z}\|_2^2$$

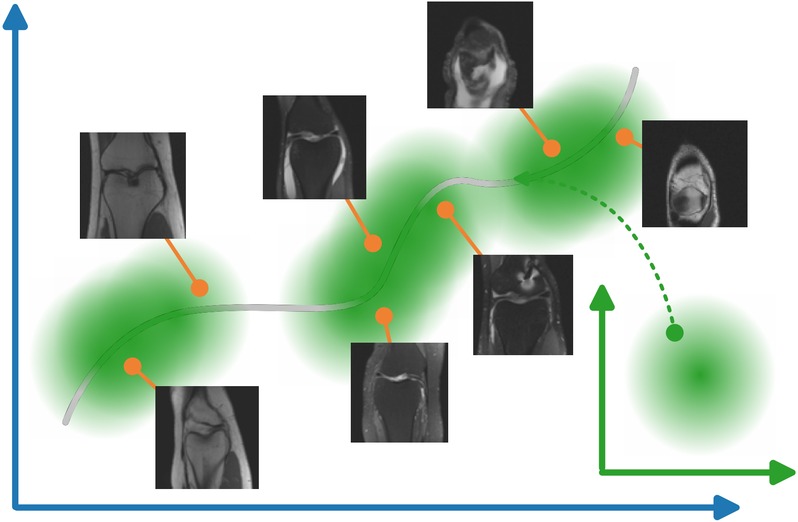
- Where the *Generator* provides $\mathcal{N}(\mathbf{x} \,|\, \mu_\theta(\mathbf{z}), \Sigma_\theta(\mathbf{z}))$ via a network $[\mu, L_\Lambda] = f(\mathbf{z}; \theta)$ and $\|\mathbf{a}\|^2_\Sigma := \mathbf{a}^\top \Sigma^{-1} \mathbf{a}$ denotes a Gaussian weighted norm
- *Note: the network still outputs $\mathcal{O}(N)$ values and evaluation of $R(\mathbf{x})$ can be performed in $\mathcal{O}(N)$ time using L_Λ for the first two terms*



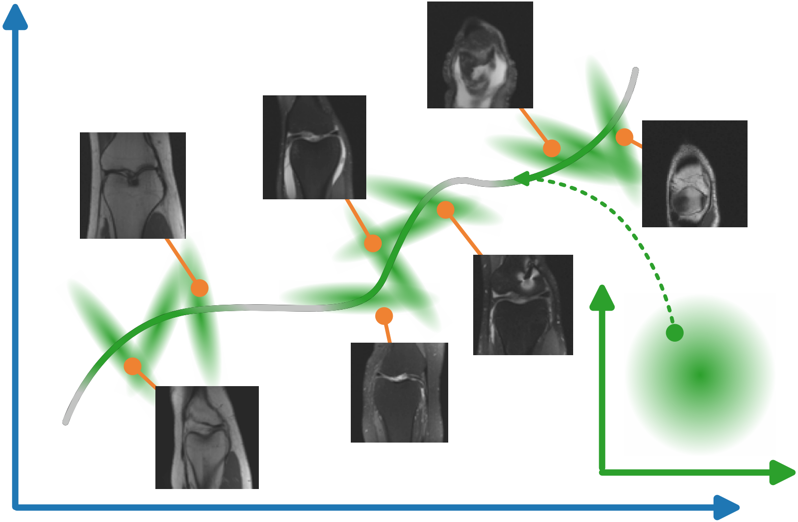
Aside: Images and manifolds



Aside: Images and manifolds



Aside: Images and manifolds



Proof of concept example: NYU fastMRI knee dataset

- Images from sampled magnitude volumes (not proper MRI!)
- Task inspired by the single-coil reconstruction
- Sample with a varying number of radial spokes
- Generator trained in two stages, first the mean, then the Cholesky
- Initialise with $\mathbf{z}^{(0)}$ using the encoding of a rough reconstruction, given by the adjoint of the forward operator, and the corresponding mean output for $\mathbf{x}^{(0)}$
- Use alternating gradient descent for $\mathbf{x}$ and $\mathbf{z}$ with backtracking line search

FastMRI knee covariance models...

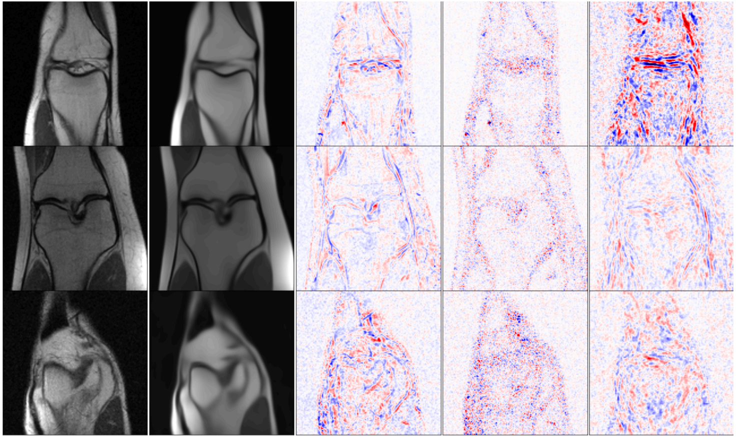


Figure 13: Samples from trained generative models with diagonal and structured covariances

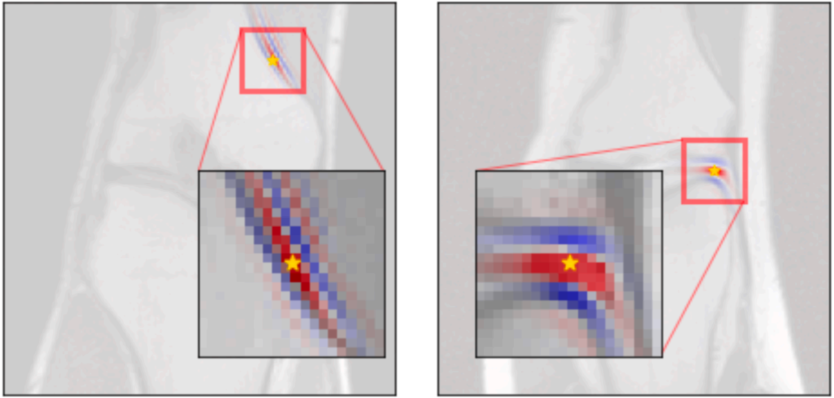


Figure 14: Visualisation of learned covariances; red indicates a high positive correlation, and blue is a strong negative correlation.

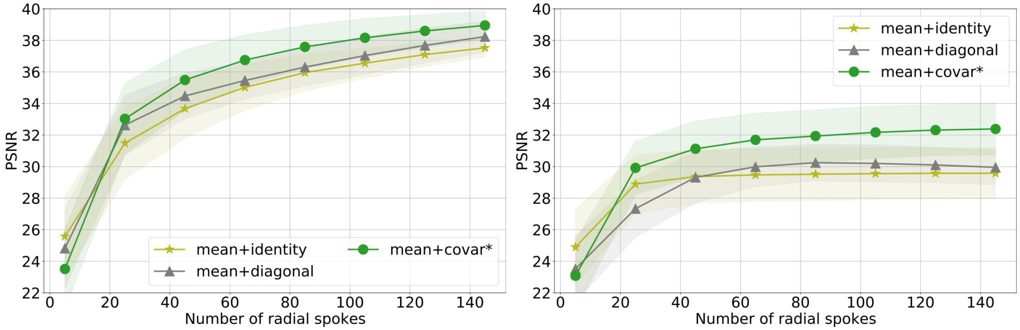


Figure 15: PSNR vs number of radial spokes. The test data was corrupted with additive Gaussian noise of standard deviation 0.0125 on the left and 0.05 on the right.

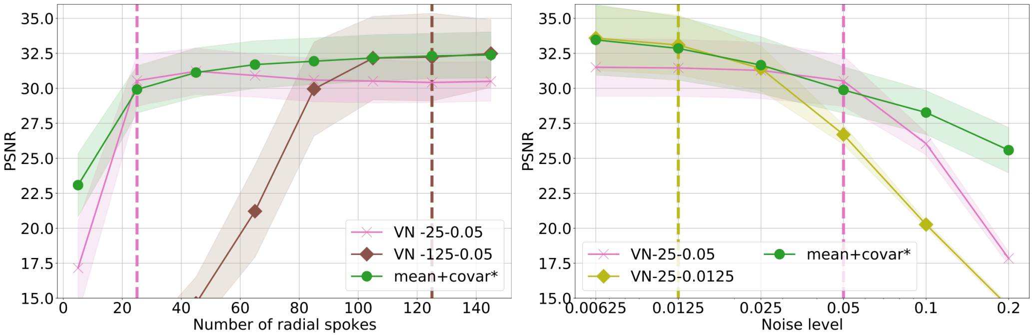


Figure 16: Comparison with the supervised variational networks [Hammernik et al. 2018]. The vertical lines depict the experimental settings the variational networks were trained on.

Comparison with optimisation of weights at test time

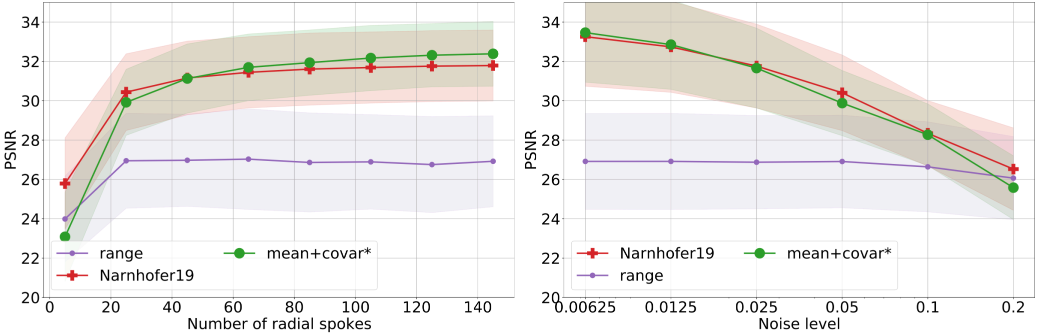


Figure 17: Comparison with constraint to the range (e.g. [Bora et al. 2017]) and optimising the generator during reconstruction [Narnhofer et al. 2019]

Example reconstruction comparison (varying number of spokes)

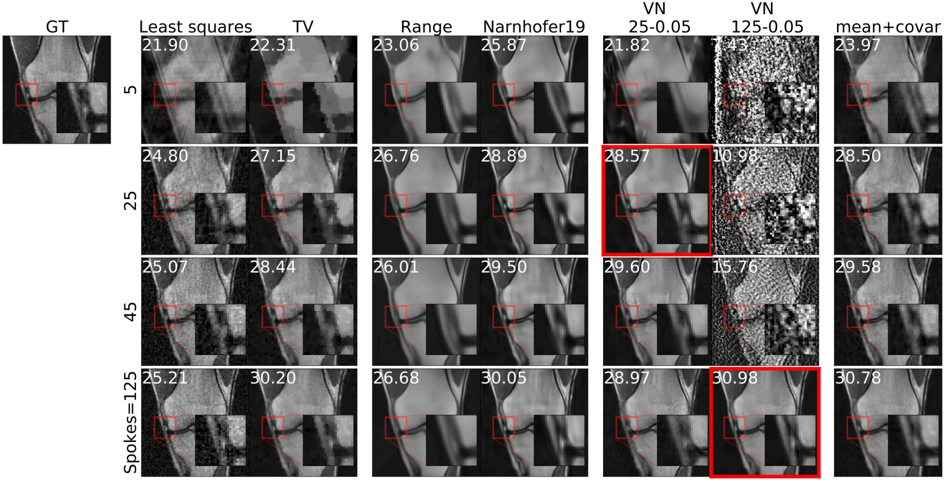


Figure 18: Varying number of spokes. The PSNR values are added in white and the red boxes indicate the settings the highlighted variational network has been trained on.

Example reconstruction comparison (varying noise)

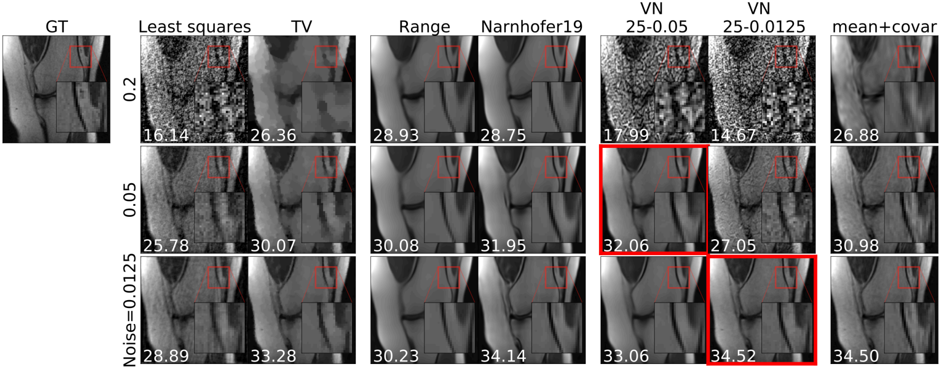


Figure 19: Varying the additive noise. The PSNR values are added in white and the red boxes indicate the settings the highlighted variational network has been trained on.

Non-Gaussian likelihoods

“Learning Structured Gaussians to Approximate Deep Ensembles”

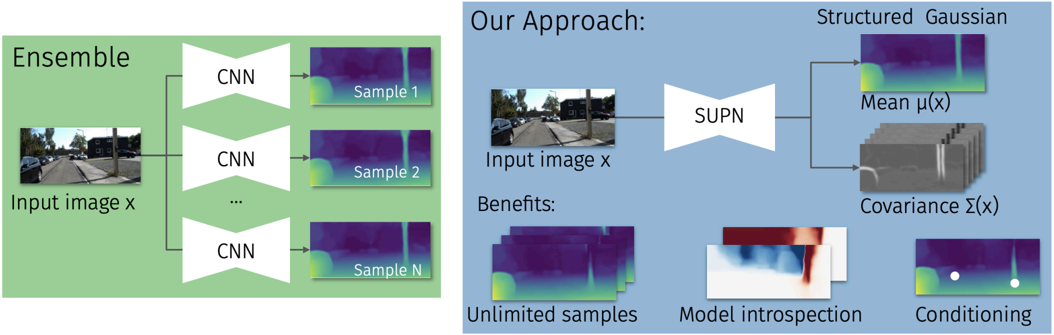


Figure 20: Use the structured Gaussian approach for “ensemble distillation”; approximate the output from a deep ensemble [Poggi et al. 2020, Lakshminarayanan et al. 2017]

[Simpson et al. 2022]

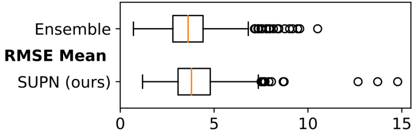
Non-Gaussian likelihood

- Use a link function to change to different likelihood (e.g. a depth range through logits)
- Training from ensemble data using log-likelihood for multiple outputs from the same input
- The output distribution is seeking to capture epistemic and aleatoric uncertainty (through the ensemble samples)

Advantages

- Efficiency improvement
- Ability to draw unlimited samples
- Introspection
- Conditional sampling

Accuracy Comparison:
The approximation captures the original ensemble well



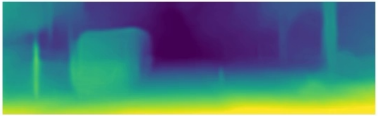
Uncertainty Metrics:
Pixelwise Area Under the Sparsification Error, Area Under the Random Gain and the Log-Likelihood

Model name	RMSE AUSE ↓	RMSE AURG ↑	LL $\times 10^5$ ↑
Ensemble [6]	2.927 (1.327)	0.324 (1.019)	
Diagonal	5.075 (1.924)	-1.697 (0.799)	1.77 (11.48)
SUPN	1.555 (1.307)	1.856 (1.355)	40.60 (1.35)

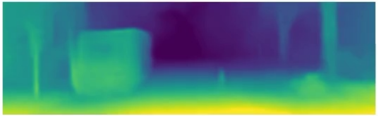
Figure 21: Monocular depth estimation results vs the original ensemble

Samples (video)

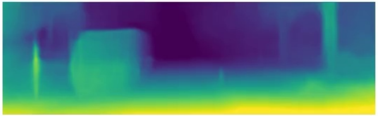
SUPN mean



Ensemble samples

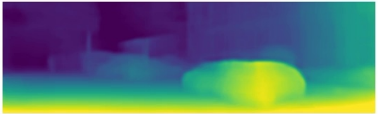


SUPN samples

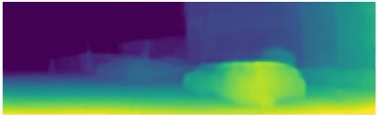


Samples (video)

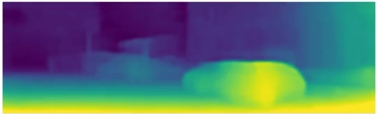
SUPN mean



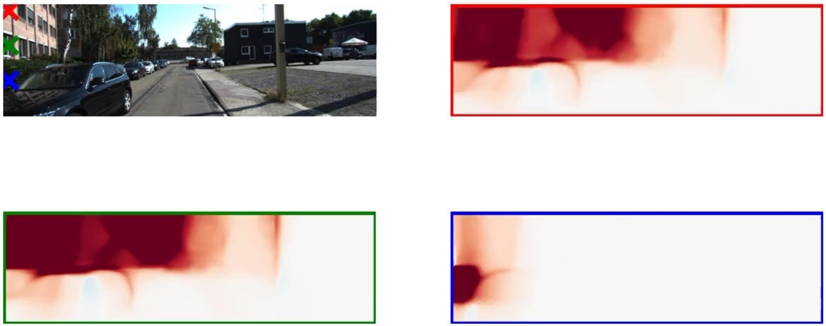
Ensemble samples



SUPN samples



Introspection (video)



Conditional sampling

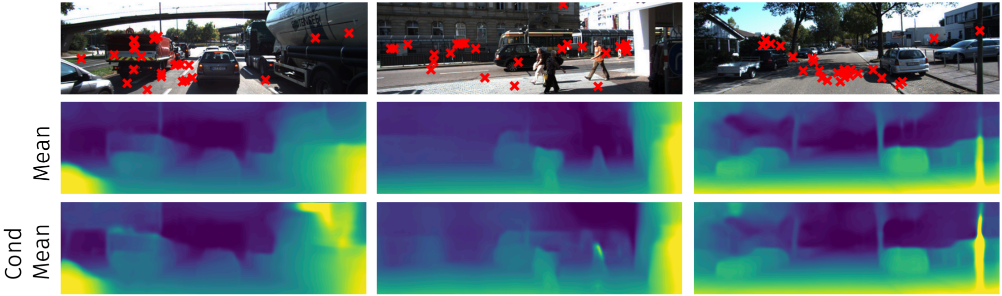


Figure 22: We can also perform *conditional* sampling using efficient sparse precision operations

$$p(\mathbf{d}_{\mathsf{U}} \,|\, \mathbf{d}_{\mathsf{K}} = \boldsymbol{\alpha}) \sim \mathcal{N}(\mathbf{b}, B)$$

$$\mathbf{b} := \boldsymbol{\mu}_{\mathsf{U}} - \Lambda_{\mathsf{UU}}^{-1} \Lambda_{\mathsf{UK}} (\boldsymbol{\alpha} - \boldsymbol{\mu}_{\mathsf{K}}), B := \Lambda_{\mathsf{UU}}^{-1}$$

Where to next?

Open challenges

- Nice introspection but what about dataset bias?
- Extensions to complex variants (e.g. proper MRI)
- Convergence rates (e.g. looking at natural gradients)
- Convexity/uniqueness
- Assumption that “ground truth” data available

Joint work with Era Dorta, Margaret Duff, Ivor Simpson, Sara Vicente, Lourdes Agapito, and Matthias Ehrhardt. Acknowledgements to the EPSRC CAMERA Research Centre, the Centre for Digital Entertainment and SAMBa CDTs, and the Royal Society.

- **Compressed Sensing MRI Reconstruction Regularized by VAEs with Structured Image Covariance,** *Margaret Duff, Ivor Simpson, Matthias J. Ehrhardt and Neill D. F. Campbell, arXiv e-print*
- **Regularising Inverse Problems with Generative Machine Learning Models,** *Margaret Duff, Neill D. F. Campbell and Matthias J. Ehrhardt, arXiv e-print*
- **Learning Structured Gaussians to Approximate Deep Ensembles,** *Ivor Simpson, Sara Vicente and Neill D. F. Campbell,* CVPR, 2022
- **Structured Uncertainty Prediction Networks,** *Era Dorta Perez, Sara Vicente, Lourdes Agapito, Neill D. F. Campbell and Ivor Simpson,* CVPR, 2018