Structured Generative Models as Priors for Inverse Problems

Neill Campbell

Joint work with Era Dorta, Margaret Duff, Ivan Ustyuzhaninov, Ieva Kazlauskaite, Markus Kaiser, Erik Bodin, Olga Mikheeva, Ivor Simpson, Sara Vicente, Lourdes Agapito, Matthias Ehrhardt, Tony Shardlow, and Carl Henrik Ek

Department of Computer Science, University of Bath

Here be monsters...


THIS IS THE WAY
makeameme.org

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

Unsupervised learning $\rightarrow$ generative models

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

Unsupervised learning $\rightarrow$ generative models

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

Unsupervised learning $\rightarrow$ generative models

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, $R(\mathbf{x}) := \log p(\mathbf{x}\,|\,\boldsymbol{\theta})$

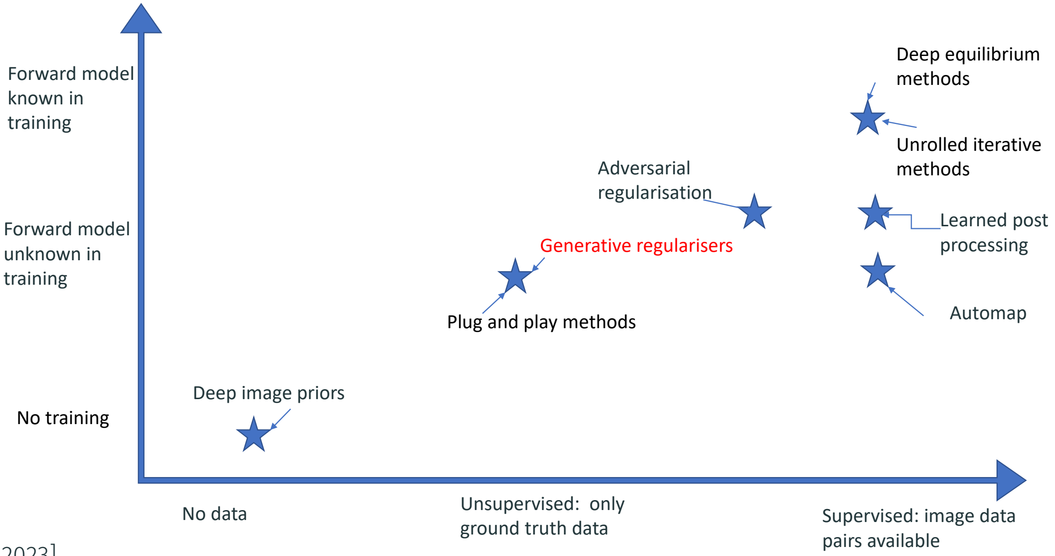
Inverse problem setup

- 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 $D(\mathbf{y}, A\,\mathbf{x}) := \log p(\mathbf{y}\,|\,f(A\,\mathbf{x}), \ldots)$

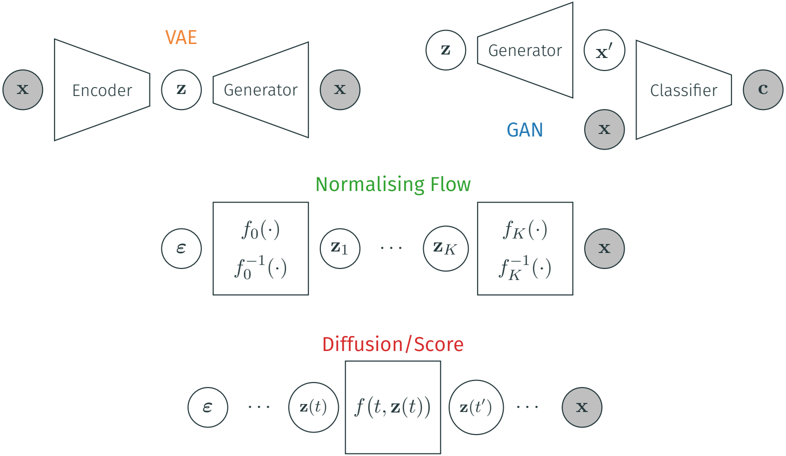
Deep learning approaches for inverse problems



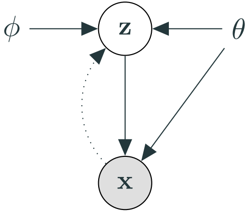
[Duff 2023]

Generative models

Generative model zoo



Unreasonable expectations of generative models?



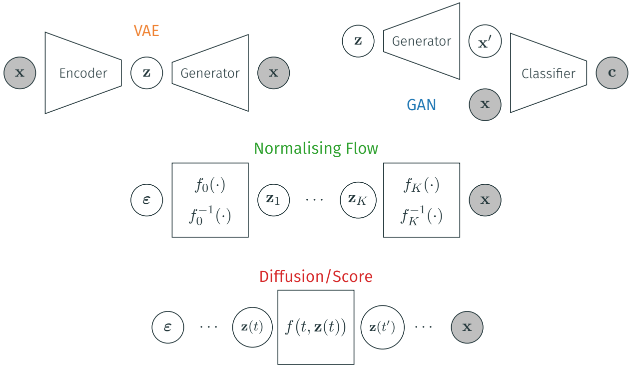
e.g. VAE with:

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

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

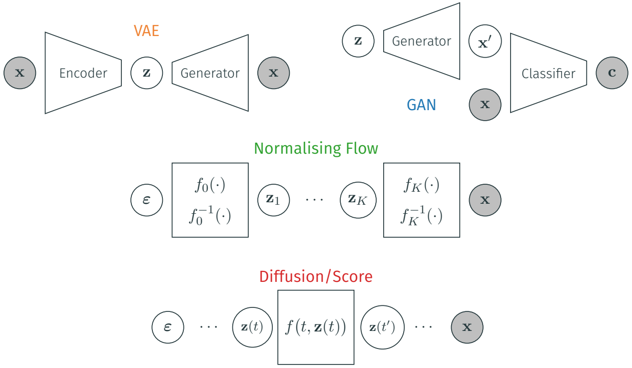
Properties we would like

- Span the data space



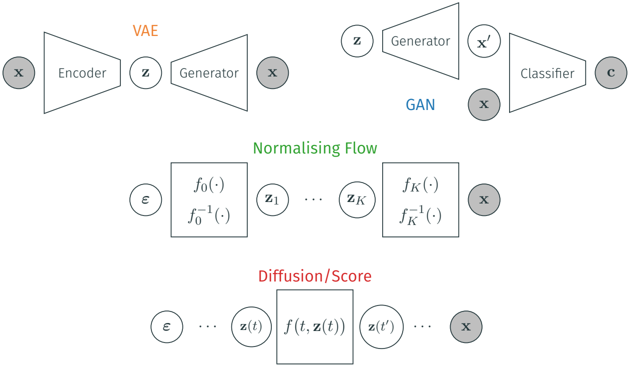
Properties we would like

- Span the data space
- Representative samples



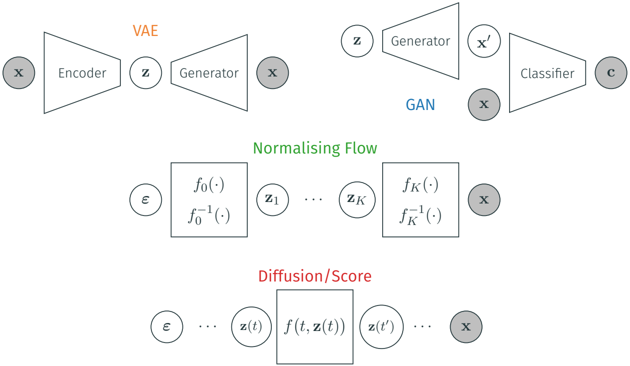
Properties we would like

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



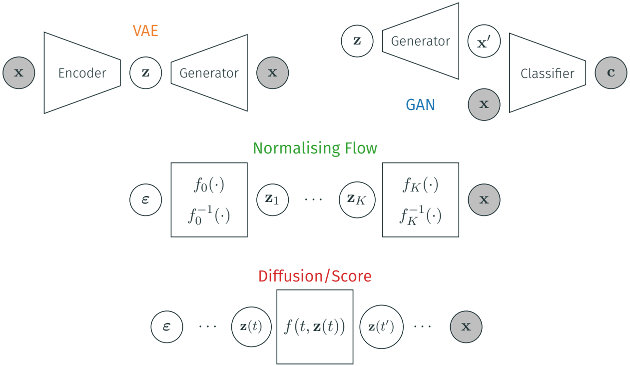
Properties we would like

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



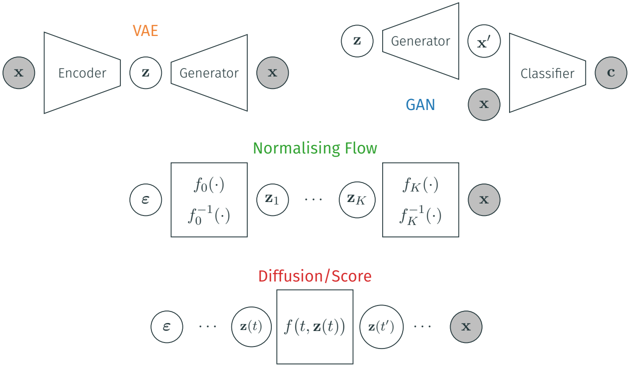
Properties we would like

- 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)

Overview...

Is unsupervised learning a thing?

Generative models

Structured Uncertainty Prediction Networks (SUPN)

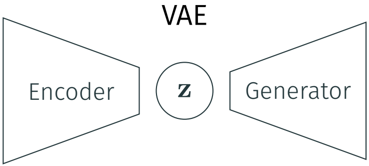
SUPN as a prior for inverse problems

Compositional Models

Where to next?

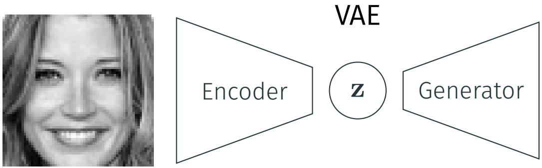
Thanks!

"VAEs produce overly smooth output"



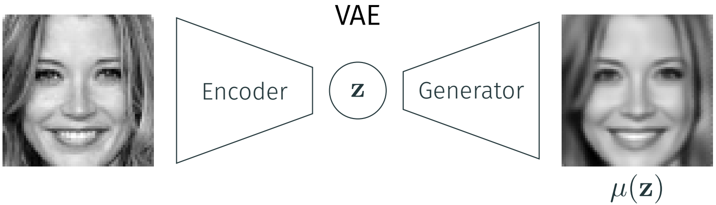
[Dorta et al. 2018]

“VAEs produce overly smooth output”



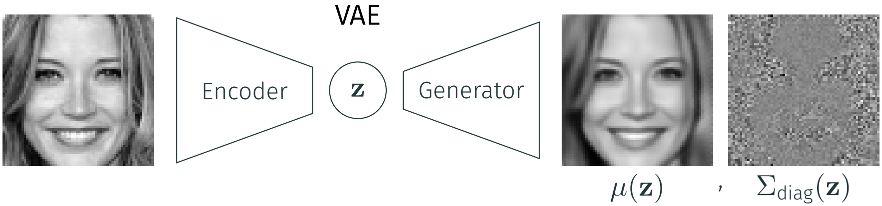
[Dorta et al. 2018]

"VAEs produce overly smooth output"



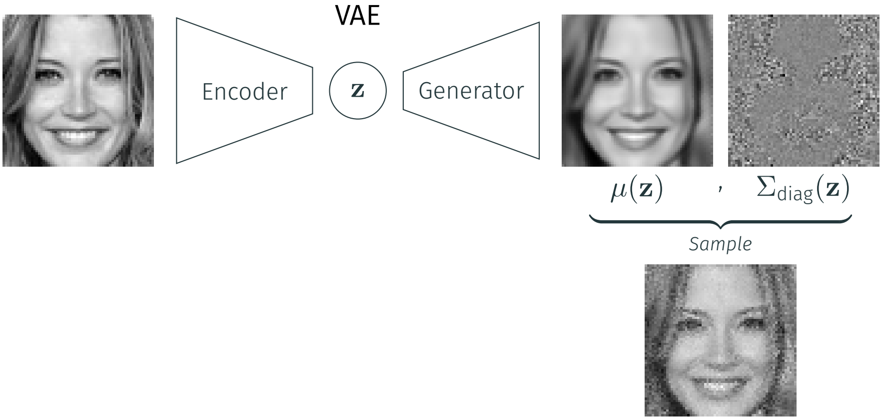
[Dorta et al. 2018]

“VAEs produce overly smooth output”



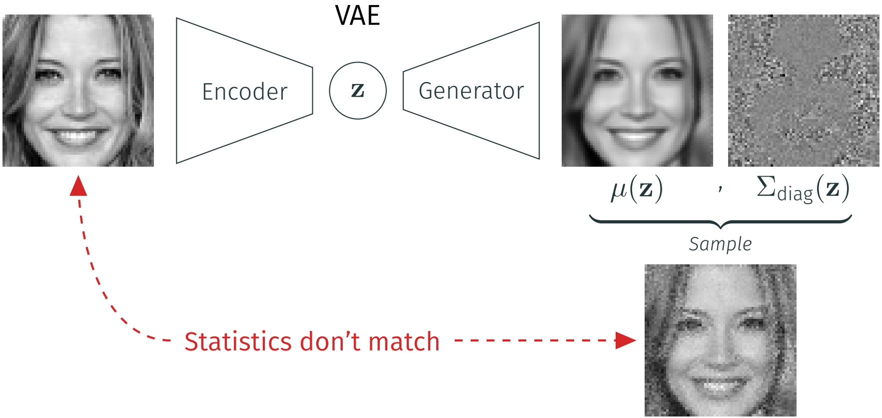
[Dorta et al. 2018]

“VAEs produce overly smooth output”



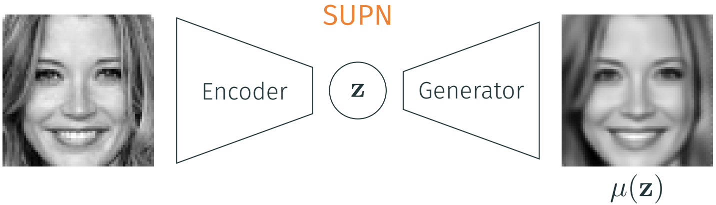
[Dorta et al. 2018]

“VAEs produce overly smooth output”



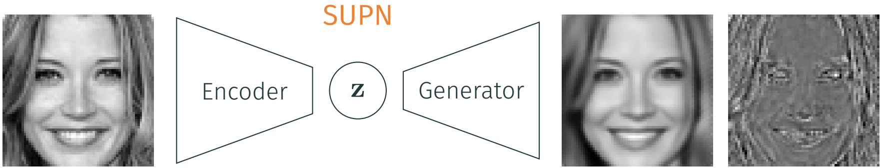
[Dorta et al. 2018]

“VAEs produce overly smooth output”



[Dorta et al. 2018]

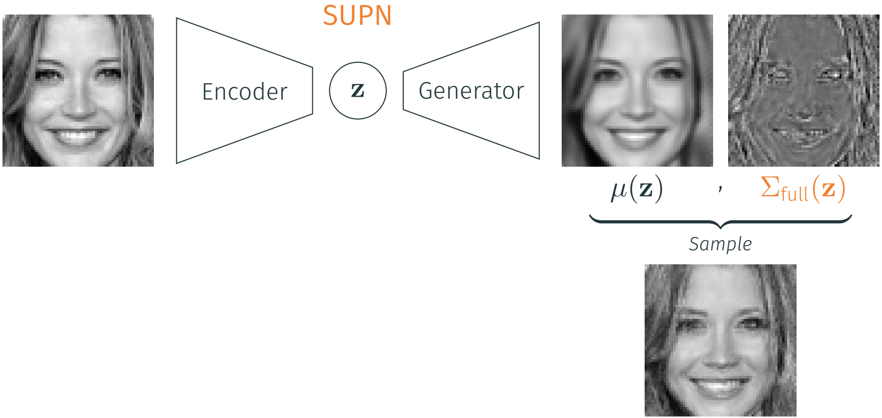
"VAEs produce overly smooth output"



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

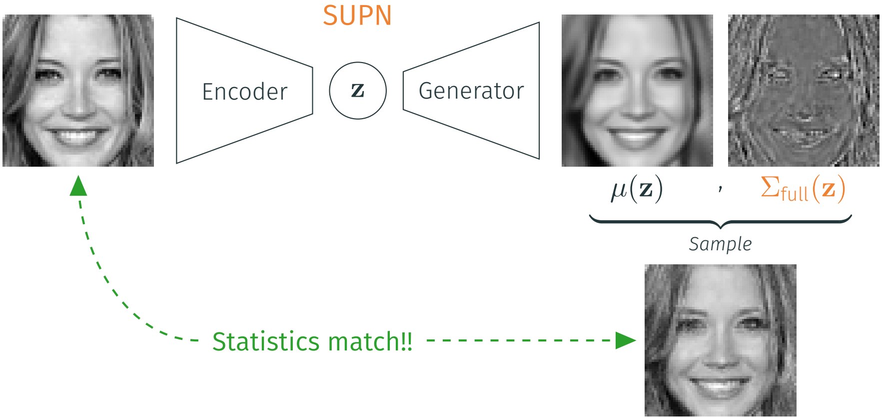
[Dorta et al. 2018]

“VAEs produce overly smooth output”



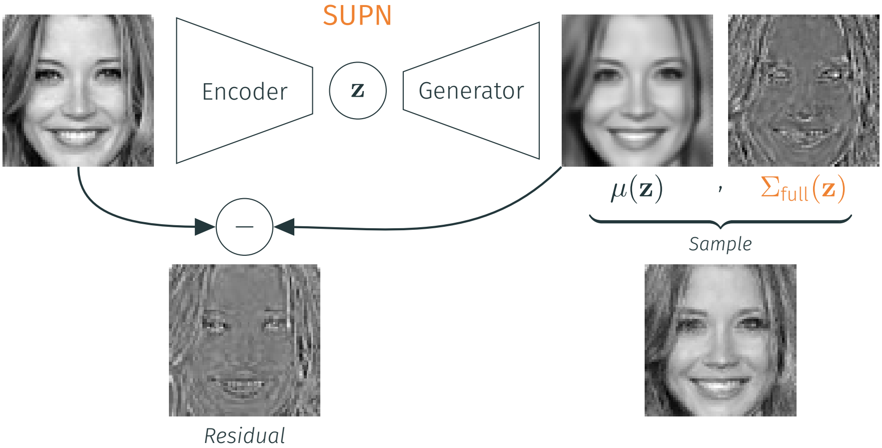
[Dorta et al. 2018]

"VAEs produce overly smooth output"



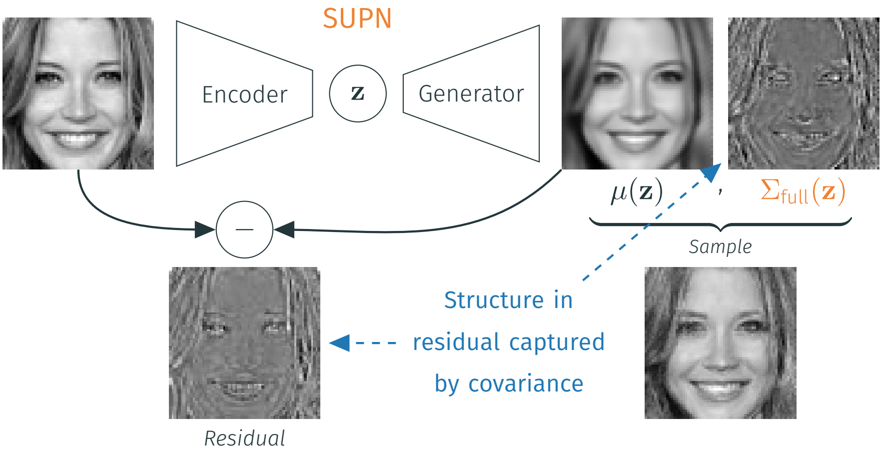
[Dorta et al. 2018]

“VAEs produce overly smooth output”



[Dorta et al. 2018]

"VAEs produce overly smooth output"



[Dorta et al. 2018]

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

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

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

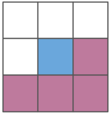
- **Problem**: $\Sigma_{\text{full}}(\mathbf{z})$ is quadratic in the number of pixels
- **Solution**: Sparse parameterisation of the Cholesky factor of the precision

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

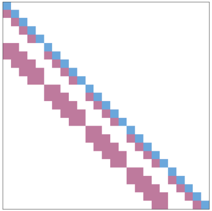
Problem! Dense covariance $\mathcal{O}(N^2)$...

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

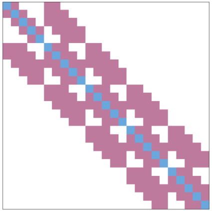
$$\Sigma(\mathbf{z}) := \left[\Lambda(\mathbf{z})\right]^{-1} := \left[L_\Lambda(\mathbf{z})\, L_\Lambda^\top(\mathbf{z})\right]^{-1}$$



Neighbourhood in image domain



Sparsity in the precision Cholesky matrix L_Λ



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

- Sparse parameterisation of the Cholesky factor of the precision

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

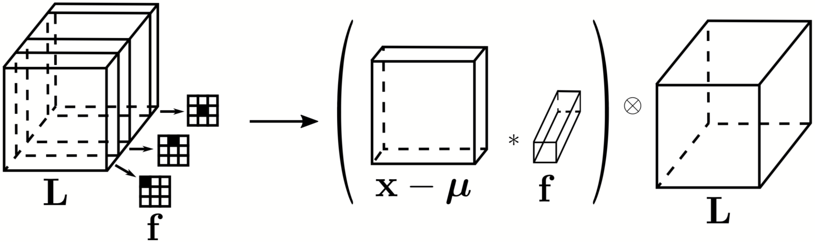


Figure 4: Implementation through convolutional structure: matrix-vector product in $\mathcal{O}(N)$

Examples of samples

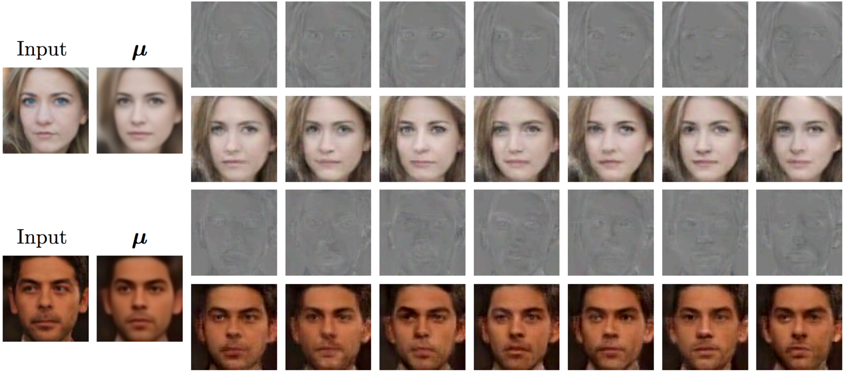


Figure 5: Variation in samples from the model on test data

Introspection of the captured covariance structure

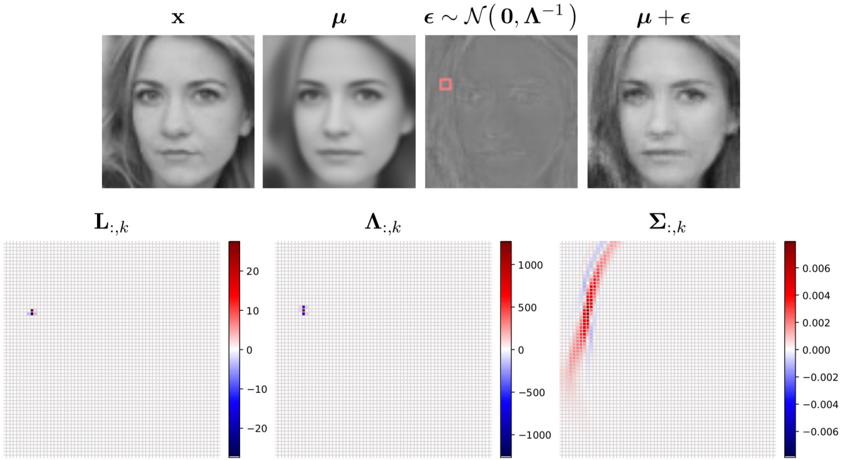


Figure 6: Visualisation of the learned correlations

Links to established concepts…

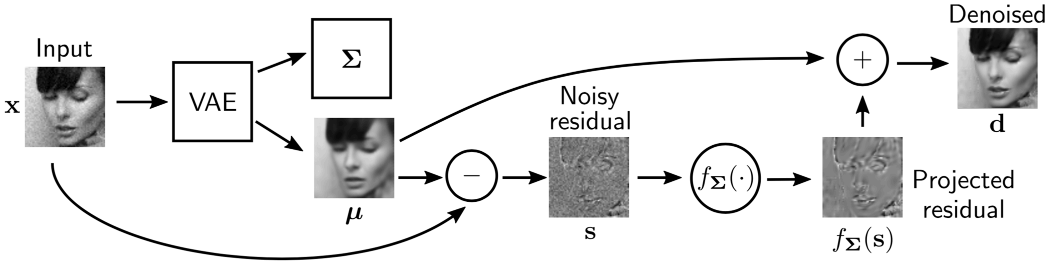
- Links to Conditional Random Field (CRF) models
  - a Gaussian CRF - e.g. “Regression Tree Fields” [Jancsary et al. 2012]
- Links to adaptive local regularisation models
  - e.g. locally adaptive TV or Laplacian based methods
- Links to Wavelet approaches
  - considering hierarchical extensions or combining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

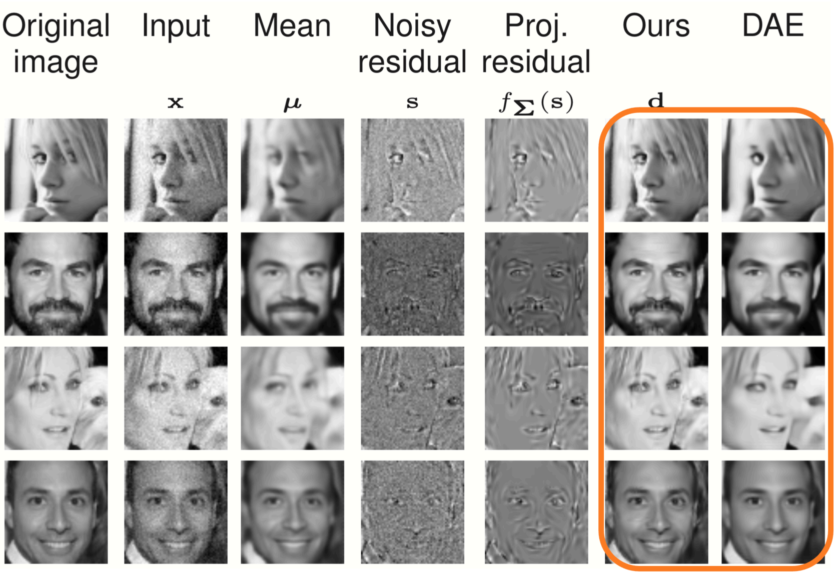
Testing with denoising...



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

Overview...

Is unsupervised learning a thing?

Generative models

Structured Uncertainty Prediction Networks (SUPN)

SUPN as a prior for inverse problems

Compositional Models

Where to next?

Thanks!

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) and $p_{\mathcal{Z}}(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, I)$

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

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

- 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

SUPN as a prior for inverse problems

- Consider a hierarchical model for the inverse problem

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

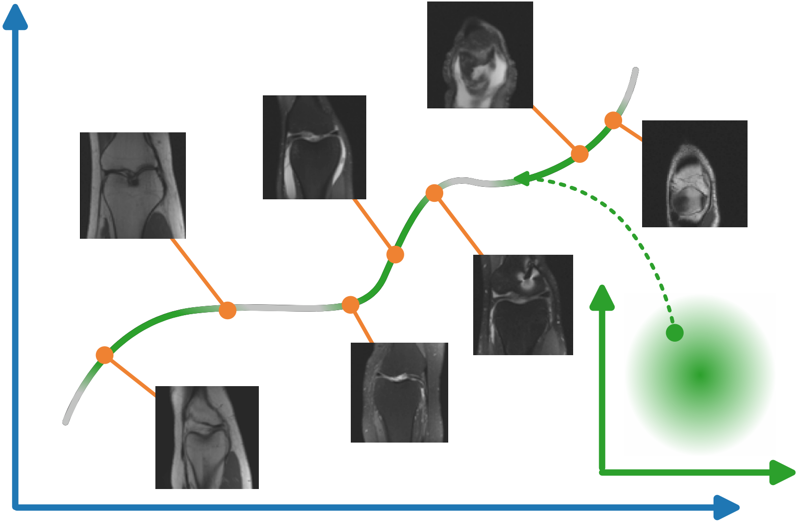
- We will take a MAP estimate for $\mathbf{z}$ rather than marginalising `:-(`
- From before (with a Gaussian observation likelihood) and $p_{\mathcal{Z}}(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, I)$

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

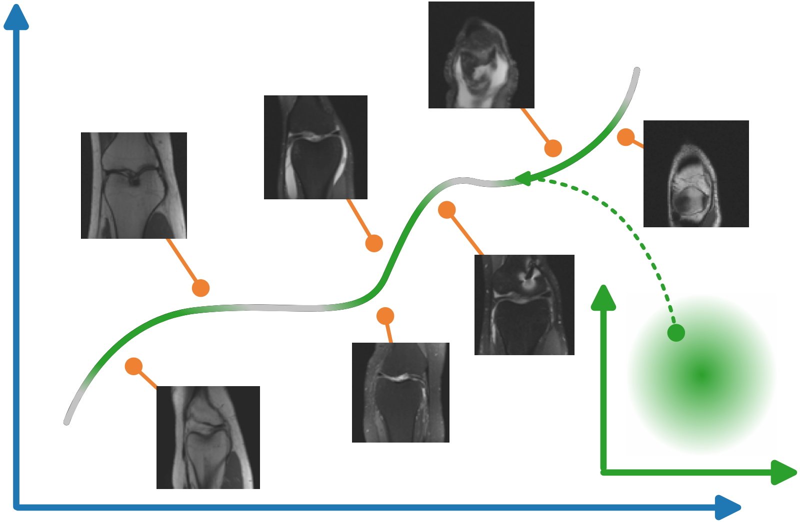
$$R(\mathbf{x}) := \min_{\mathbf{z}\in\mathcal{Z}}\ \log|\Sigma_\theta(\mathbf{z})| + \frac{1}{2}\|\mathbf{x} - \mu_\theta(\mathbf{z})\|^2_{\Sigma_\theta(\mathbf{z})} + \frac{1}{2}\|\mathbf{z}\|_2^2$$

- 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*

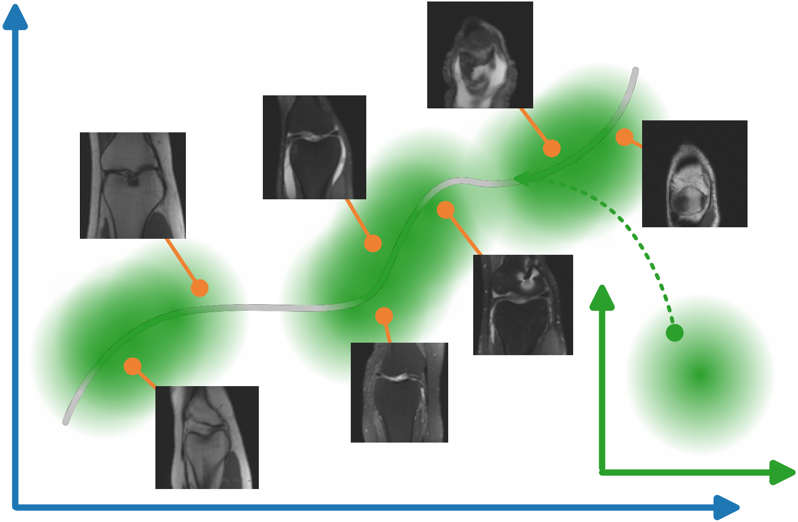
Aside: Images and manifolds



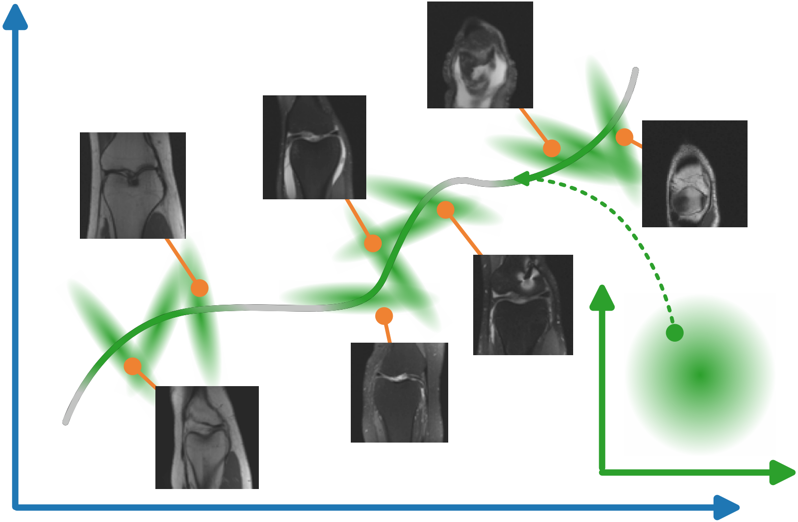
Aside: Images and manifolds



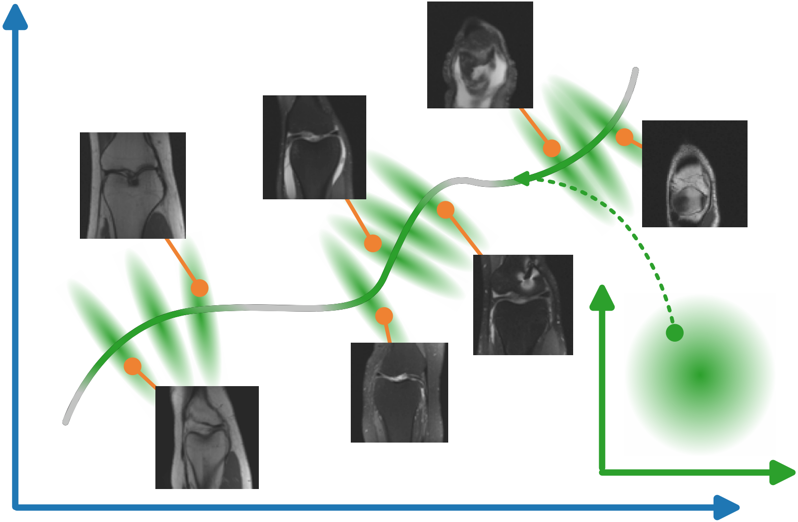
Aside: Images and manifolds



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...

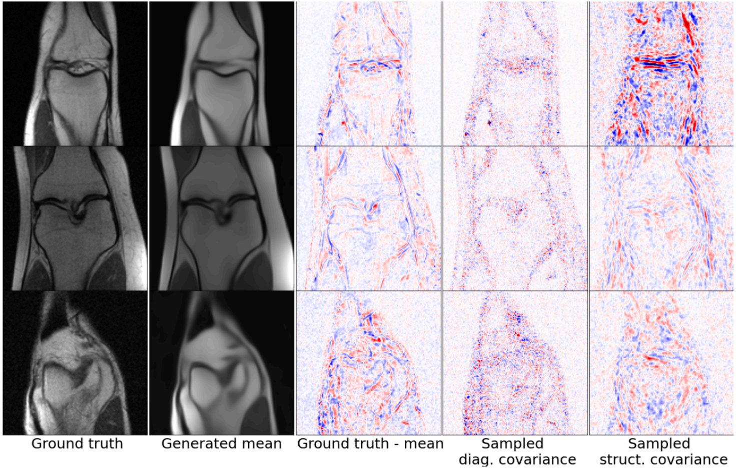


Figure 14: Samples from trained generative models with diagonal and structured covariances

Introspection: Visualisation of learned covariances...

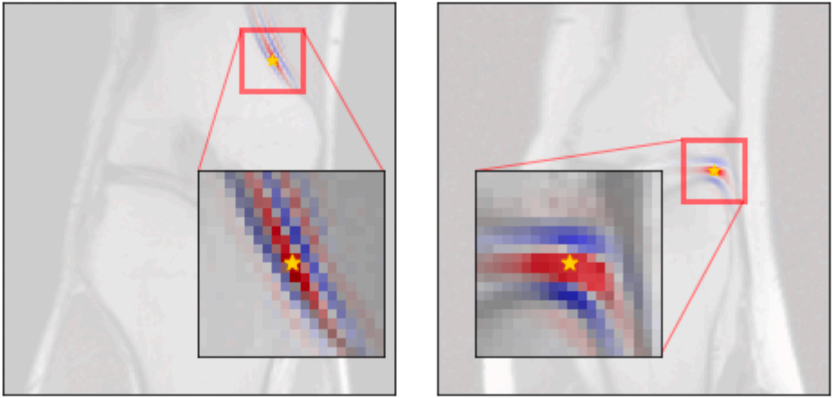


Figure 15: Visualisation of learned covariances; red indicates a high positive correlation, and blue is a strong negative correlation.

Comparison vs supervised reconstruction method

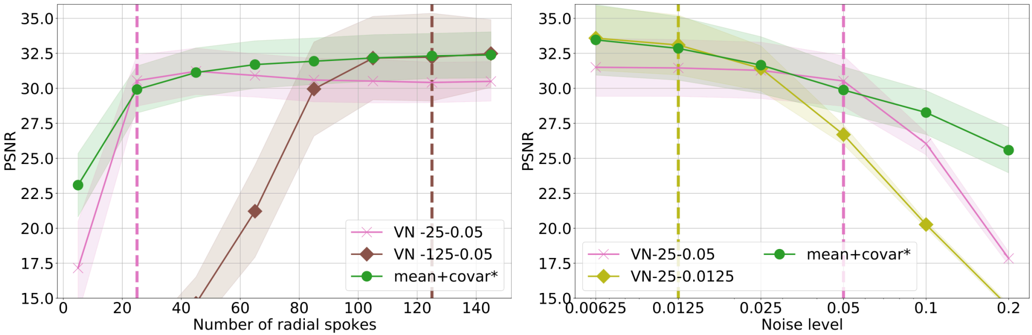


Figure 16: Comparison with the supervised variational networks [Hammernik et al. 2018]. The vertical lines depict the experimental settings the variational networks were trained on.

Example reconstruction comparison (varying number of spokes)

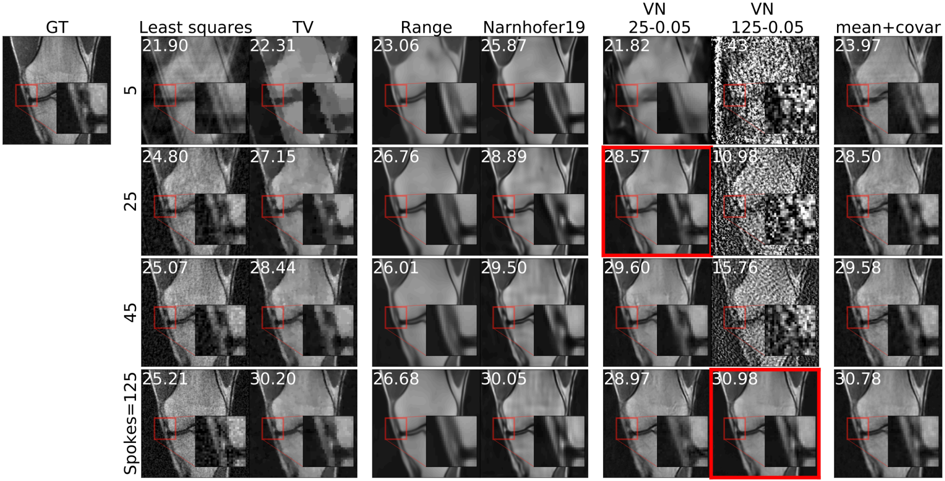


Figure 17: Varying number of spokes. The PSNR values are added in white and the red boxes indicate the settings the highlighted variational network has been trained on.

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

Dropped my Bayesian Card (tm) somewhere along the way..

3rd Workshop on Uncertainty Quantification for Computer Vision

ECCV 2024 Workshop

In the last decade, substantial progress has been made w.r.t. the performance of computer vision systems, a significant part of it thanks to deep learning. These advancements prompted sharp community growth and a rise in industrial investment. However, most current models lack the ability to reason about the confidence of their predictions; integrating uncertainty quantification into vision systems will help recognize failure scenarios and enable robust applications.

In addition to advances in Bayesian deep learning, providing practical approaches for vision problems, the workshop will provide a forum for discussing promising research directions, which have received less attention, as well as advancing current practices to drive future research. Examples include: the development of new metrics that reflect the real-world need for uncertainty when using vision systems with down-stream tasks; and moving beyond point-estimates to address the multi-modal ambiguities inherent in many vision tasks.

This years UNcertainty quantification for Computer Vision (UNCV) Workshop aims to raise awareness and generate discussion regarding how predictive uncertainty can, and should, be effectively incorporated into models within the vision community. The workshop will bring together experts from machine learning and computer vision to create a new generation of well-calibrated and effective methods that *know when they do not know*.

Compositional Models

Overview...

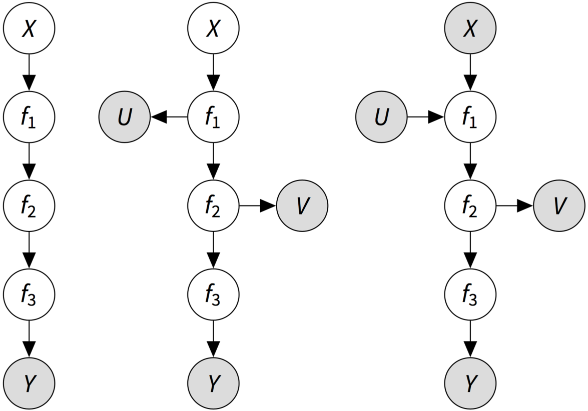


Figure 18: Examples of composite models

Compositional models

- Hierarchical/composite models
- More deep GPs than deep Bayesian Neural Networks (although some thoughts applicable)

Compositional models

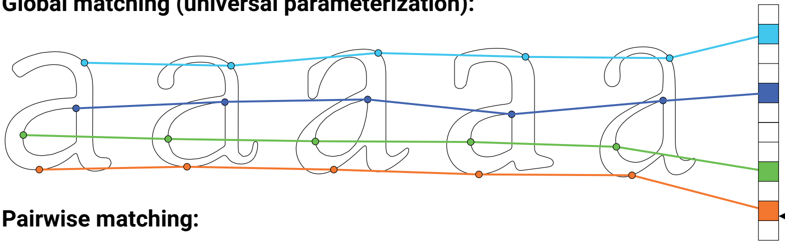
- Hierarchical/composite models
- More deep GPs than deep Bayesian Neural Networks (although some thoughts applicable)
- Such models are likely to contain “compositional uncertainty”

Compositional models

- Hierarchical/composite models
- More deep GPs than deep Bayesian Neural Networks (although some thoughts applicable)
- Such models are likely to contain "compositional uncertainty"
- Related to ideas around identifiability from statistics

Example of composition: alignment

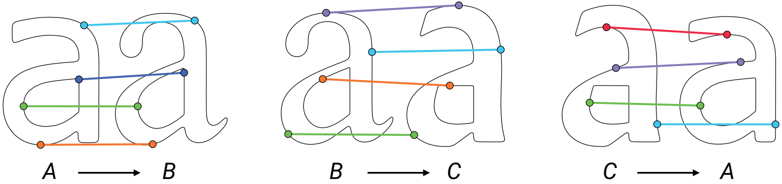
Global matching (universal parameterization):



Correspondences found globally across all examples..

..unique parameterization in **parameter vector**

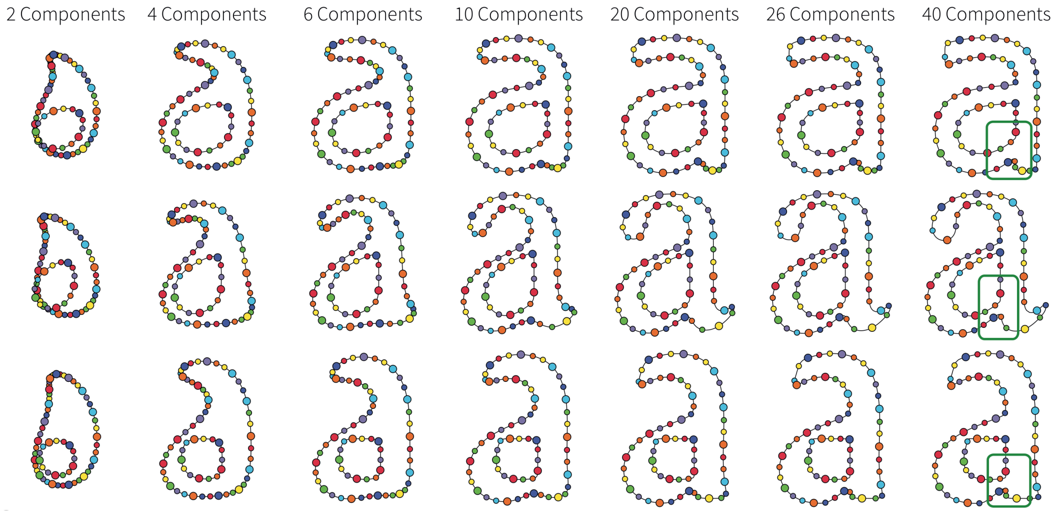
Pairwise matching:



Consistency problem:

$$A \to B \to C \to A \neq I$$

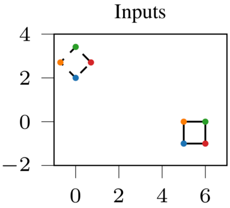
Example of composition: alignment



Uncertainty within compositions..

- Illustration: rigid shape transformation..

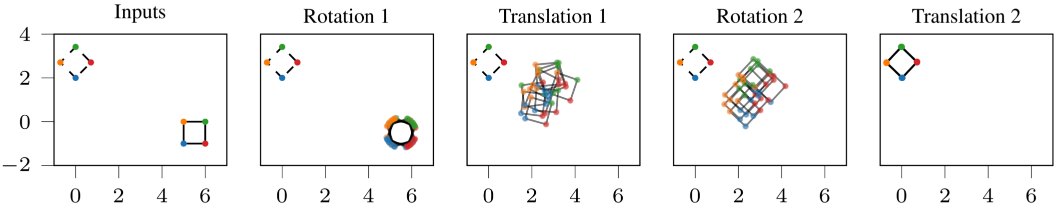
$$\text{input} \rightarrow R_1 \rightarrow T_1 \rightarrow R_2 \rightarrow T_2 \rightarrow \text{output}$$



Uncertainty within compositions..

- Illustration: rigid shape transformation..

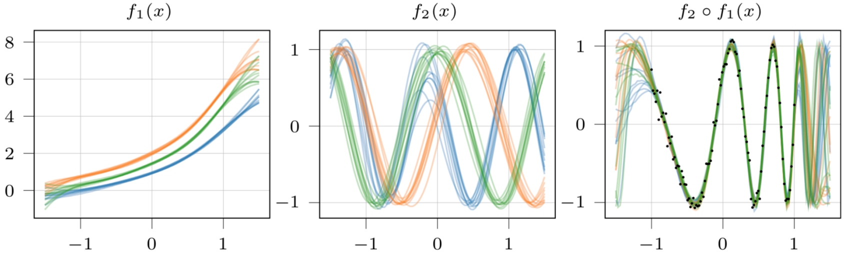
$$\text{input} \to R_1 \to T_1 \to R_2 \to T_2 \to \text{output}$$



- Here under-constrained → uncertainty

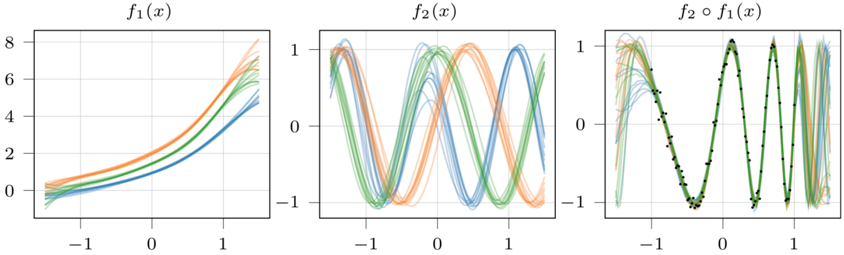
Examples..

- Two layer decomposition of a chirp:

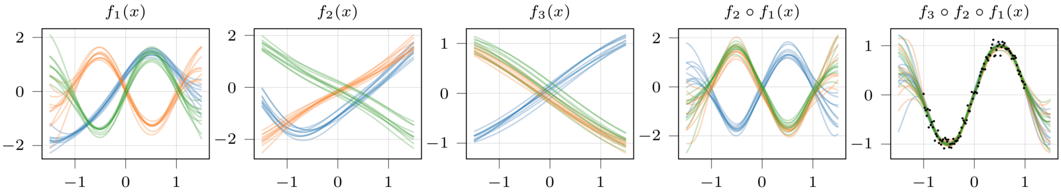


Examples..

- Two layer decomposition of a chirp:



- Three layer decomposition of a sinusoid:



Background on Hierarchical/Composite/Deep GPs..

- A deep GP is a distribution over compositions of functions

$$f = f_L \circ \ldots \circ f_1$$

  where each f_i is a regular GP

- Typically we use a formulation based on the Sparse Variational GP
- Each layer maintains a set of inducing distributions $q(U_i)$ specified at set of corresponding inducing locations
- The training goal is to approximate the posterior $p(\{U_i\} \mid Y, X)$ with these distributions

Doubly Stochastic Variational Inference (DSVI)

Variational approximation scheme [Salimbeni 2017] Given our data $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}$ *we model*

$$\mathbf{y}_n = (f_L \circ \cdots \circ f_1)(\mathbf{x}_n) + \boldsymbol{\varepsilon}_n$$

with $f_l \sim \mathcal{GP}(\mu_l(\cdot), \kappa_l(\cdot,\cdot))$ *We use* $\mathbf{F}_l \sim (f_l \circ \cdots \circ f_1)(\mathbf{X})$ *to denote the evaluation of the entire input data* $\mathbf{X}$ *at layer* $l = 2, \ldots, L$ *The joint distribution (with* $\mathbf{F}_0 := \mathbf{X}$*) is*

$$p(\mathbf{Y}, \mathbf{F}_L, \ldots, \mathbf{F}_1 \mid \mathbf{X}) = p(\mathbf{Y} \mid \mathbf{F}_L) \prod_{l=1}^{L} p(\mathbf{F}_L \mid \mathbf{F}_{l-1})$$

Importantly, we cannot perform the marginalisation integral as the Gaussian factors are contained inside non-linear kernels

We seek a lower bound

$$\mathcal{L} \leq p(\mathbf{Y}, \mathbf{F}_L, \ldots, \mathbf{F}_1 \mid \mathbf{X}) = p(\mathbf{Y} \mid \mathbf{F}_L) \prod_{l=1}^{L} p(\mathbf{F}_L \mid \mathbf{F}_{l-1})$$

We define inducing locations $\{\mathbf{Z}_l\}$ *and function output* $\{\mathbf{U}_l\}$ *for each layer*

$$p(\mathbf{Y}, \{\mathbf{F}_l\}, \{\mathbf{U}_l\} \mid \mathbf{X}, \{\mathbf{Z}_l\}) = p(\mathbf{Y} \mid \mathbf{F}_L) \prod_{l=1}^{L} p(\mathbf{F}_L \mid \mathbf{F}_{l-1}, \mathbf{U}_l, \mathbf{Z}_{l-1})\, p(\mathbf{U}_l \mid \mathbf{Z}_{l-1})$$

There is a specific form for the GP posteriors $p(\mathbf{F}_L \mid \mathbf{F}_{l-1}, \mathbf{U}_l, \mathbf{Z}_{l-1}) \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$

$$\boldsymbol{\mu}_l = \mu_l(\mathbf{F}_{l-1}) + \alpha_l(\mathbf{F}_{l-1})^\top (\mathbf{U}_l - \mu_l(\mathbf{F}_{l-1}))$$

$$\boldsymbol{\Sigma}_l = \kappa_l(\mathbf{F}_{l-1}, \mathbf{F}_{l-1}) - \alpha_l(\mathbf{F}_{l-1})^\top \kappa_l(\mathbf{Z}_{l-1}, \mathbf{Z}_{l-1})\, \alpha_l(\mathbf{F}_{l-1})$$

where $\alpha_l(\mathbf{F}_{l-1}) := [\kappa_l(\mathbf{Z}_{l-1}, \mathbf{Z}_{l-1})]^{-1} \kappa_l(\mathbf{Z}_{l-1}, \mathbf{F}_{l-1})$

The factorised variational distributions are then introduced

$$q(\{\mathbf{U}_l\}) = q(\mathbf{U}_1)\dots q(\mathbf{U}_L),\; q(\mathbf{U}_l) \sim \mathcal{N}(\mathbf{m}_l, \mathbf{S}_l)$$

The lower bound is then

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{F}_L)}\left[\log p(\mathbf{Y} \mid \mathbf{F}_L)\right] - \sum_{l=1}^{L} \mathrm{KL}\left[q(\mathbf{U}_l) \| p(\mathbf{U}_l \mid \mathbf{Z}_{l-1})\right]$$

The key DSVI insight is an efficient MC estimation of the expectation by marginalising the inducing points $\{\mathbf{U}_l\}$ *from the variational posterior*

$$\begin{aligned} q(\{\mathbf{F}_l\}) &= \prod_{l=1}^{L} \int p(\mathbf{F}_l \mid \mathbf{U}_l)\, q(\mathbf{U}_l)\, \mathrm{d}\mathbf{U}_l \\ &= q(\mathbf{F}_L \mid \mathbf{F}_{L-1})\dots q(\mathbf{F}_1 \mid \mathbf{X}),\; \textit{with } q(\mathbf{F}_l \mid \mathbf{F}_{l-1}) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}) \end{aligned}$$

$$\begin{aligned} \textit{where } \tilde{\boldsymbol{\mu}} &:= \mu_l(\mathbf{F}_{l-1}) + \alpha_l(\mathbf{F}_{l-1})^\top (\mathbf{m}_l - \mu_l(\mathbf{F}_{l-1})) \\ \tilde{\boldsymbol{\Sigma}} &:= \kappa_l(\mathbf{F}_{l-1}, \mathbf{F}_{l-1}) - \alpha_l(\mathbf{F}_{l-1})^\top \left[\kappa_l(\mathbf{Z}_{l-1}, \mathbf{Z}_{l-1}) - \mathbf{S}_l\right] \alpha_l(\mathbf{F}_{l-1}) \end{aligned}$$

Problem with mean field inference...

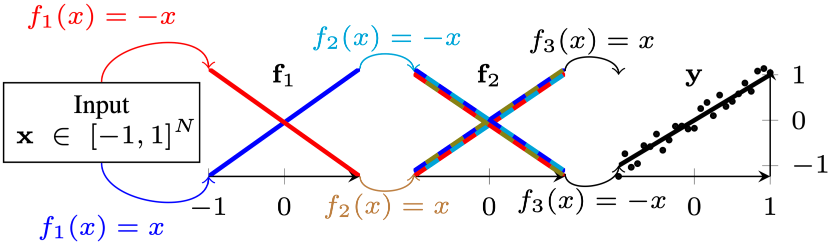
- Issue with the mean field assumption (i.e. each layer modelled independently)

$$q(\{\mathbf{U}_l\}) = q(\mathbf{U}_1) \dots q(\mathbf{U}_L),\ q(\mathbf{U}_l) \sim \mathcal{N}(\mathbf{m}_l, \mathbf{S}_l)$$

Problem with mean field inference...

- Issue with the mean field assumption (i.e. each layer modelled independently)

$$q(\{\mathbf{U}_l\}) = q(\mathbf{U}_1)\ldots q(\mathbf{U}_L),\ q(\mathbf{U}_l) \sim \mathcal{N}(\mathbf{m}_l, \mathbf{S}_l)$$



Quantitative argument

- Assume DGP layers are independent $\{f_l\}$

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l - 1$ form uncertain inputs to layer l

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l-1$ form uncertain inputs to layer l
- Similar to [Mchutchon 2011] we can analyse as $f(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l-1$ form uncertain inputs to layer l
- Similar to [Mchutchon 2011] we can analyse as $f(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$
- Consider a single input x, we can write $\mathbf{F}_l = f_l(\mathbf{F}_{l-1}) = f_l(\bar{\mathbf{F}}_{l-1} + \varepsilon_{l-1})$ where $\bar{\mathbf{F}}_{l-1}$ is the mean and ε_{l-1} denotes a zero-mean distortion

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l-1$ form uncertain inputs to layer l
- Similar to [Mchutchon 2011] we can analyse as $f(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$
- Consider a single input x, we can write $\mathbf{F}_l = f_l(\mathbf{F}_{l-1}) = f_l(\bar{\mathbf{F}}_{l-1} + \varepsilon_{l-1})$ where $\bar{\mathbf{F}}_{l-1}$ is the mean and ε_{l-1} denotes a zero-mean distortion
- Note, ε_{l-1} are *not* necessarily Gaussian (as the marginals of a Deep GP are not Gaussian in general). We denote the variance as $\sigma_{\mathrm{n}}^2 := \mathbb{V}[\varepsilon_{l-1}]$

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l-1$ form uncertain inputs to layer l
- Similar to [Mchutchon 2011] we can analyse as $f(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$
- Consider a single input x, we can write $\mathbf{F}_l = f_l(\mathbf{F}_{l-1}) = f_l(\bar{\mathbf{F}}_{l-1} + \varepsilon_{l-1})$ where $\bar{\mathbf{F}}_{l-1}$ is the mean and ε_{l-1} denotes a zero-mean distortion
- Note, ε_{l-1} are *not* necessarily Gaussian (as the marginals of a Deep GP are not Gaussian in general). We denote the variance as $\sigma_{\mathrm{n}}^2 := \mathbb{V}[\varepsilon_{l-1}]$
- We want to show that the variance of $\mathbf{F}_l$ increases with increasing variance of ε_{l-1}

Quantitative argument

- Assume DGP layers are independent $\{f_l\}$
- Distribution of outputs of layer $l-1$ form uncertain inputs to layer l
- Similar to [Mchutchon 2011] we can analyse as $f(\mathbf{x}+\boldsymbol{\varepsilon}_{\mathbf{x}})$
- Consider a single input x, we can write $\mathbf{F}_l = f_l(\mathbf{F}_{l-1}) = f_l(\bar{\mathbf{F}}_{l-1} + \varepsilon_{l-1})$ where $\bar{\mathbf{F}}_{l-1}$ is the mean and ε_{l-1} denotes a zero-mean distortion
- Note, ε_{l-1} are *not* necessarily Gaussian (as the marginals of a Deep GP are not Gaussian in general). We denote the variance as $\sigma_{\mathrm{n}}^2 := \mathbb{V}[\varepsilon_{l-1}]$
- We want to show that the variance of $\mathbf{F}_l$ increases with increasing variance of ε_{l-1}
- Therefore, unless the layers *collapse*, i.e. $\varepsilon_{l-1} \to 0$, the variance at the final $\mathbf{F}_L$ will be large and a poor fit to data

Linear approximation for layers (uncertain input)

We approximate $\mathbf{F}_l = f_l(\mathbf{F}_{l-1}) \approx f_l(\bar{\mathbf{F}}_l) + \varepsilon_{l-1} f_l'(\bar{\mathbf{F}}_l)$ *where* $f_l(\bar{\mathbf{F}}_l) \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_l, \bar{\boldsymbol{\sigma}}_l^2)$ *(both functions of* $\bar{\mathbf{F}}_{l-1}$*) Recalling that a GP and its derivative are jointly distributed*

$$\begin{bmatrix} f_l(\bar{\mathbf{F}}_l) \\ f_l'(\bar{\mathbf{F}}_l) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_l \\ \boldsymbol{\mu}_l' \end{bmatrix}, \begin{bmatrix} \bar{\boldsymbol{\sigma}}_l^2 & (\bar{\boldsymbol{\sigma}}_l^2)' \\ (\bar{\boldsymbol{\sigma}}_l^2)' & (\bar{\boldsymbol{\sigma}}_l^2)'' \end{bmatrix} \right)$$

Computing a linear transform we have

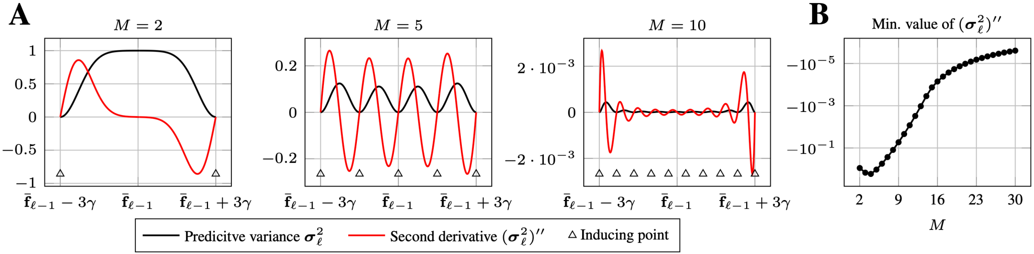
$$\mathbb{E}[\mathbf{F}_l \mid \varepsilon_{l-1}] = \bar{\boldsymbol{\mu}}_l + \varepsilon_{l-1} \bar{\boldsymbol{\mu}}_l'$$
$$\mathbb{V}[\mathbf{F}_l \mid \varepsilon_{l-1}] = \bar{\boldsymbol{\sigma}}_l^2 - 2\,\varepsilon_{l-1} (\bar{\boldsymbol{\sigma}}_l^2)' + \varepsilon_{l-1}^2 (\bar{\boldsymbol{\sigma}}_l^2)''$$

Using the law of total variance we have

$$\begin{aligned} \mathbb{V}[\mathbf{F}_l] &= \mathbb{E}[\mathbb{V}[\mathbf{F}_l \mid \varepsilon_{l-1}]] + \mathbb{V}[\mathbb{E}[\mathbf{F}_l \mid \varepsilon_{l-1}]] \\ &= \bar{\boldsymbol{\sigma}}_l^2 + \sigma_{\mathrm{n}}^2 \left[(\bar{\boldsymbol{\mu}}_l')^2 + (\bar{\boldsymbol{\sigma}}_l^2)'' \right] + \mathcal{O}(\varepsilon_{l-1}^2) \end{aligned}$$

Illustration of posterior variance

- The only term that can be negative for $\mathbb{V}[\mathbf{F}_l]$ is $(\bar{\boldsymbol{\sigma}}_l^2)''$
- Illustration with M linearly spaced inducing points over a range $\Delta_l := [\bar{\mathbf{F}}_{l-1} - 3\gamma_l, \bar{\mathbf{F}}_{l-1} + 3\gamma_l]$ where γ_l is the kernel lengthscale for layer l.



- Minimum of $(\bar{\boldsymbol{\sigma}}_l^2)'' \to 0$ as M increases; a negative value indicates all inducing points are far from $\bar{\mathbf{F}}_{l-1}$; this would imply a poor data fit

How do we fix this?

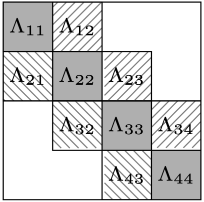
1. Jointly Gaussian variational distribution

$$q(\mathbf{U}_1, \ldots, \mathbf{U}_L) \sim \mathcal{N}(\mathbf{m}, \mathbf{S}), \; \mathbf{m} \in \mathbb{R}^{LM}, \; \mathbf{S} \in \mathbb{R}^{LM \times LM}$$

   but both expensive and tricky to evaluate

- Can make progress with a chain-like factorisation

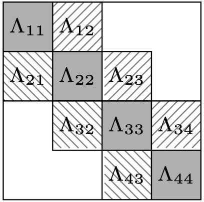
$$q(\{\mathbf{U}_l\}) = q(\mathbf{U}_L \mid \mathbf{U}_{L-1}) \ldots q(\mathbf{U}_2 \mid \mathbf{U}_1) \, q(\mathbf{U}_1)$$



How do we fix this?

1. Jointly Gaussian variational distribution

$$q(\mathbf{U}_1, \ldots, \mathbf{U}_L) \sim \mathcal{N}(\mathbf{m}, \mathbf{S}), \ \ \mathbf{m} \in \mathbb{R}^{LM}, \ \ \mathbf{S} \in \mathbb{R}^{LM \times LM}$$



but both expensive and tricky to evaluate

- Can make progress with a chain-like factorisation

$$q(\{\mathbf{U}_l\}) = q(\mathbf{U}_L \mid \mathbf{U}_{L-1}) \ldots q(\mathbf{U}_2 \mid \mathbf{U}_1)\, q(\mathbf{U}_1)$$

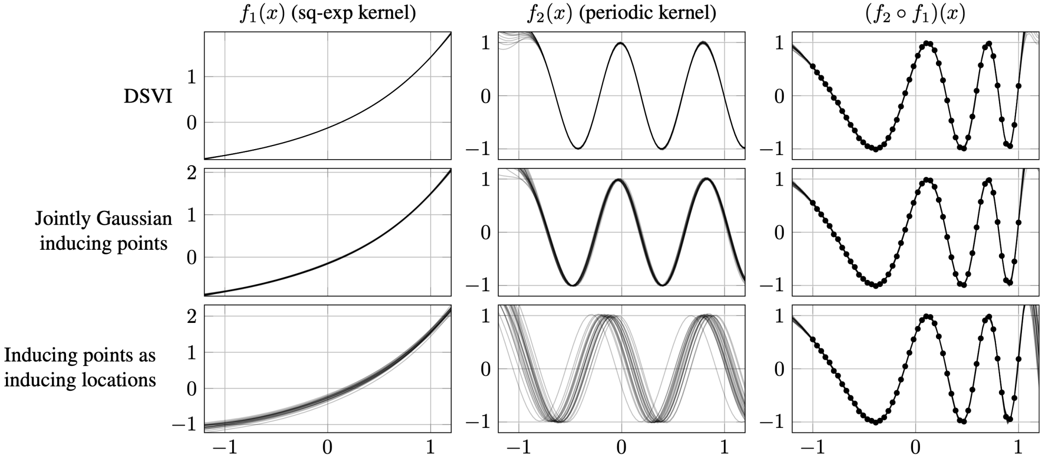
2. Inducing points as inducing locations; that is $\mathbf{U}_l \to \mathbf{F}_l^{\mathbf{Z}} \sim (f_l \circ \cdots \circ f_1)(\mathbf{Z})$

- Thus the inducing outputs of the previous layer are the inducing locations for the next

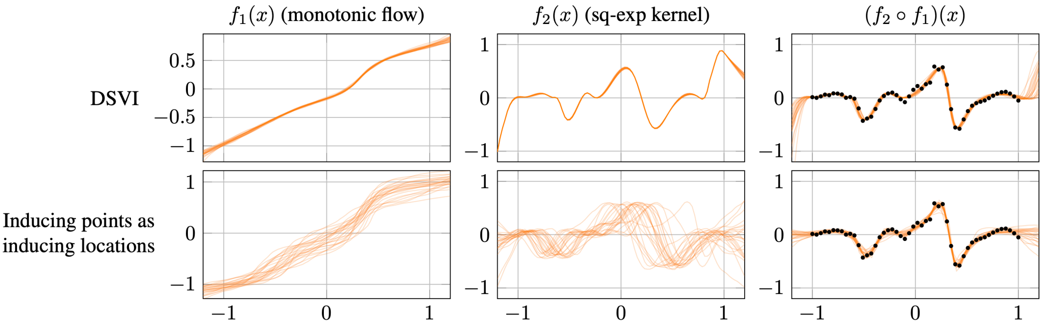
$$\mathcal{L} := \mathbb{E}_{q(\mathbf{F}_L)}\left[\log p(\mathbf{Y} \mid \mathbf{F}_L)\right] - \sum_{l=1}^{L} \mathbb{E}_{q(\mathbf{F}_l^{\mathbf{Z}})q(\mathbf{F}_{l-1}^{\mathbf{Z}})}\left[\log \frac{q(\mathbf{F}_l^{\mathbf{Z}})}{p(\mathbf{F}_l^{\mathbf{Z}} \mid \mathbf{F}_{l-1}^{\mathbf{Z}})}\right]$$

- Efficient estimation procedure in $\mathcal{O}(LNM^3)$

Fitting a chirp signal (changing lengthscale)

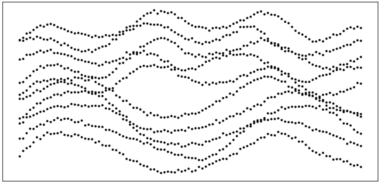


Fitting a compositional model to heartbeat data

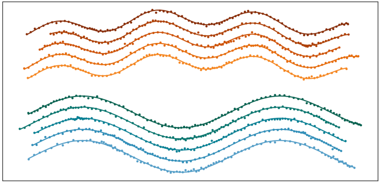


Application: Alignments

- **Multi-task Learning**: misalignment hinders ability to learn correct correlations between tasks



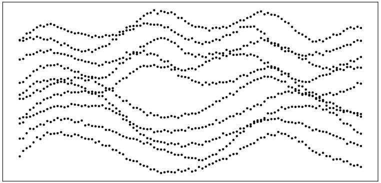
Observed data



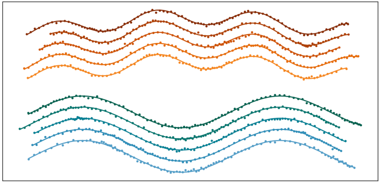
Aligned data

Application: Alignments

- **Multi-task Learning**: misalignment hinders ability to learn correct correlations between tasks
- Previous approaches:



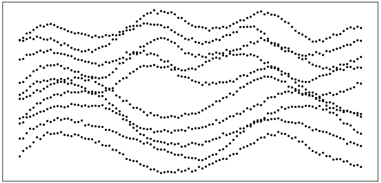
Observed data



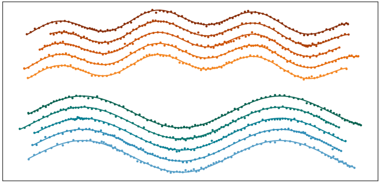
Aligned data

Application: Alignments

- **Multi-task Learning**: misalignment hinders ability to learn correct correlations between tasks
- Previous approaches:
  - Only model fixed alignment



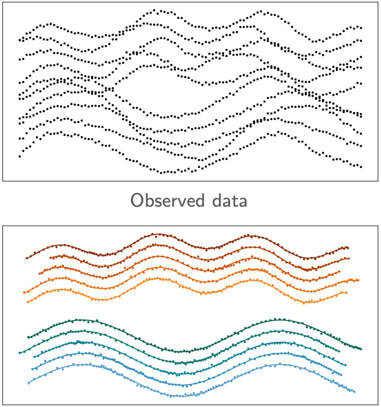
Observed data



Aligned data

Application: Alignments

- **Multi-task Learning**: misalignment hinders ability to learn correct correlations between tasks
- Previous approaches:
  - Only model fixed alignment
  - a-priori knowledge of task correlations

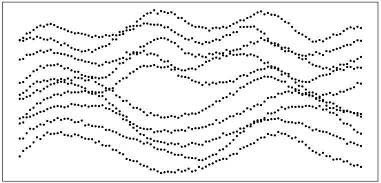


Observed data

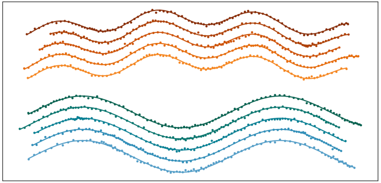
Aligned data

Application: Alignments

- **Multi-task Learning**: misalignment hinders ability to learn correct correlations between tasks
- Previous approaches:
  - Only model fixed alignment
  - a-priori knowledge of task correlations
  - either probabilistic or monotonic alignment but not both



Observed data



Aligned data

Monotonic process for temporal alignment

- Temporal warping must not permute time
- Compromises required for existing monotonic GPs
- Propose ODE-based Monotonic GP Flow

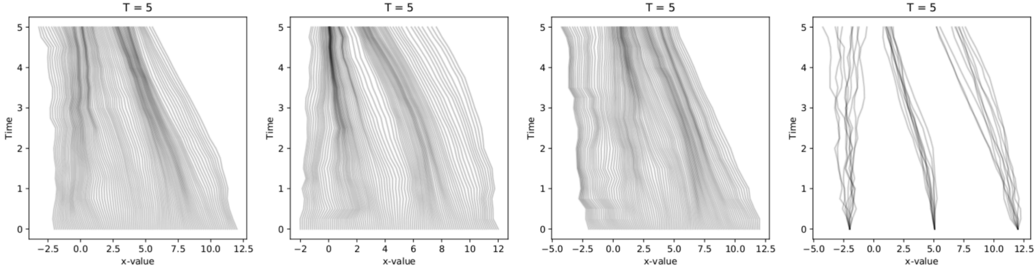
$$g(x) := u(\tau = T; x) = \int_0^T w(u(\tau))\, \mathrm{d}\tau$$

$$\text{ODE: } \mathrm{d}u = w(u)\, \mathrm{d}\tau,$$

$$\text{Uncertain drift function: } w(u) \sim \mathcal{GP}(\mathbf{0}, \kappa_w(u, u))$$

- ODE solution $g(x)$ is monotonic wrt the initial condition $u(\tau = 0) := x$
- Efficient path-wise GP sampling to solve [Terenin 2021]

Monotonic process intuition...



Monotonic process intuition...

Aligned Multi-Task GP

Our model: Fully Bayesian multi-task learning for misaligned data

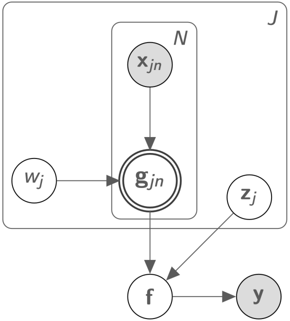
Latent corr. $\mathbf{z}_j \sim \mathcal{N}(\mathbf{z}_j \mid \mathbf{0}, \mathbf{I}_Q)$

ODE Drift $w_j \sim \mathcal{GP}\left(w_j \mid \mathbf{0}, K_\omega(u_j, u_j)\right)$

Warp $\mathbf{g}_j \mid \mathbf{x}_j, w_j \sim \text{Monotonic Process}(\mathbf{g}_j \mid \mathbf{x}_j, w_j)$

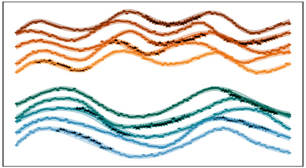
Function $\mathbf{f} \mid \mathbf{z}, \mathbf{g} \sim \mathcal{GP}\left(\mathbf{f} \mid \mathbf{0}, K_\psi(\mathbf{z}_j, \mathbf{z}_{j'}) \odot K_\theta(\mathbf{g}_{j,n}, \mathbf{g}_{j',n'})\right)$

Noisy data $\mathbf{y} \mid \mathbf{f} \sim \mathcal{N}(\mathbf{y} \mid \mathbf{f}, \beta^{-1}\mathbf{I}_{JN})$

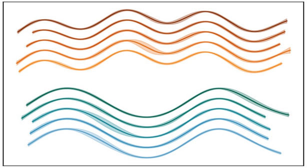


Joint prob.: $$p(\mathbf{y}, \mathbf{f}, \mathbf{z}, \mathbf{g}, w \mid \mathbf{X}) = p(\mathbf{f} \mid \mathbf{z}, \mathbf{g}) \prod_{j=1}^{J} p(\mathbf{g}_j \mid \mathbf{x}_j, w_j)\, p(w_j)\, p(\mathbf{z}_j) \prod_{n=1}^{N} p(y_{jn} \mid f_{jn})$$

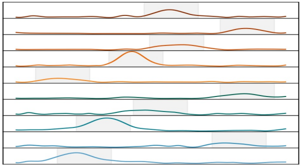
Aligned Multi-Task GP



(a) Observations and data fit

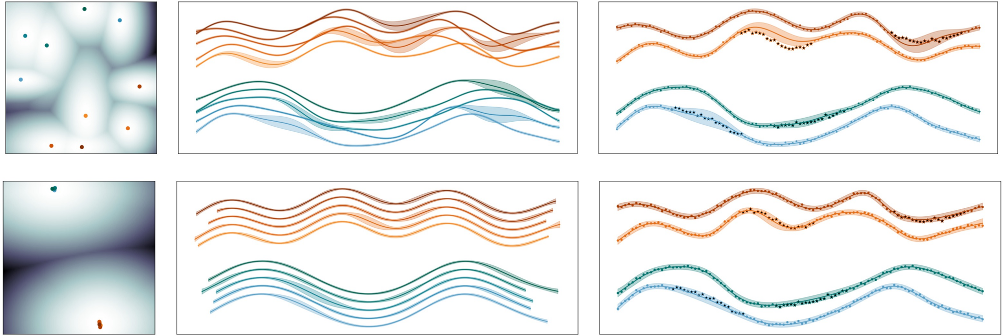


(b) Aligned multi-task GP



(c) Uncertainty in the warps

Aligned Multi-Task GP



Where to next?

Overview...

Is unsupervised learning a thing?

Generative models

Structured Uncertainty Prediction Networks (SUPN)

SUPN as a prior for inverse problems

Compositional Models

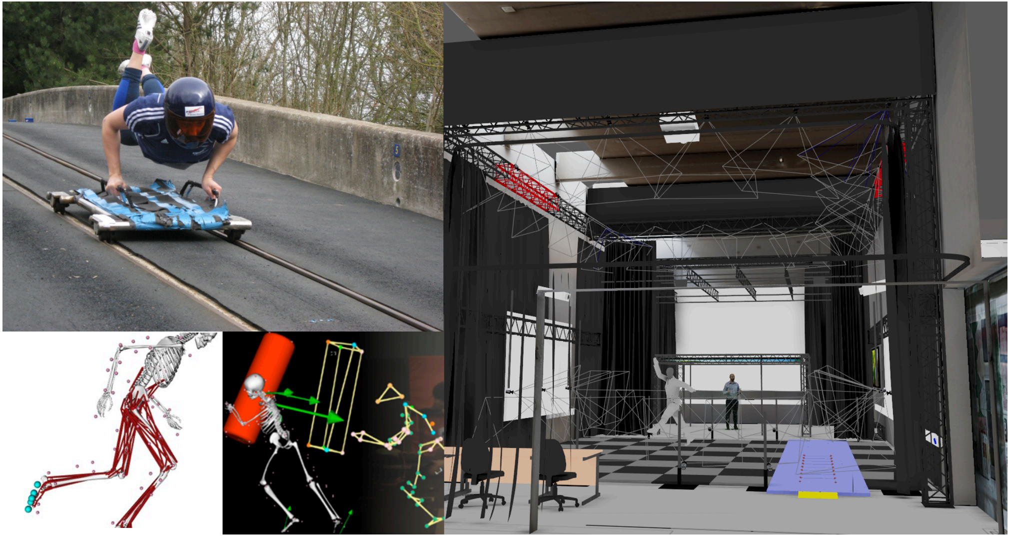
Where to next?

Thanks!

Centre for the Analysis of Motion, Entertainment Research and Applications

CAMERA

Understanding (human) motion...Would like to talk more!



Thanks!

Thanks! https://www.ndfcampbell.org n.campbell@bath.ac.uk

Joint work with Era Dorta, Margaret Duff, Ivan Ustyuzhaninov, Ieva Kazlauskaite, Markus Kaiser, Erik Bodin, Ivor Simpson, Sara Vicente, Lourdes Agapito, Matthias Ehrhardt, Tony Shardlow, and Carl Henrik Ek. Thanks 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
- **Aligned Multi-Task Gaussian Process**, *Olga Mikheeva, Ieva Kazlauskaite, Adam Hartshorne, Hedvig Kjellström, Carl Henrik Ek and Neill D. F. Campbell*, AISTATS, 2022
- **Compositional Uncertainty in Deep Gaussian Processes**, *Ivan Ustyuzhaninov, Ieva Kazlauskaite, Markus Kaiser, Erik Bodin, Neill D. F. Campbell and Carl Henrik Ek*, UAI, 2020
- **Structured Uncertainty Prediction Networks,** *Era Dorta, Sara Vicente, Lourdes Agapito, Neill D. F. Campbell and Ivor Simpson*, CVPR, 2018