Data Assimilation as Variational Inference
Full posterior estimation using the 4DVAR cost

Arthur Filoche[1] and Dominique Béréziat[2]

[1] University of Western Australia (Perth, Australia)
[2] Sorbonne Université, LIP6 (Paris, France)

Advancements in Variational Data Assimilation

Motivation

Research Interest:

- ▷ intersection of **Data Assimilation** and **Machine Learning**
- ▷ optimizing models **directly** on **imperfect** geo-scientific **observations**
- ▷ **4DVAR**: physics-based regularizer in the form of a dynamical model

Today's Topic

Linking Data Assimilation and Variational Inference

Outline

I. Data Assimilation as Variational Inference

- Variational Data Assimilation
- Variational inference
- Full posterior estimation using the 4DVAR cost

II. Case study on Lorenz96 model

- Twin experiment
- Results

III. Perspectives

- Normalizing Flow
- Amortized Inference
- Accounting for model error

I. Data Assimilation as Variational Inference

Variational Data Assimilation

Data Assimilation framework

- System state: $\mathbf{X}_t$
- Dynamics: $\mathbf{X}_{t+1} = \mathbb{M}(\mathbf{X}_t)$ *perfect model hypothesis*
- Observations: $\mathbf{Y}_t = \mathbb{H}_t(\mathbf{X}_t) + \varepsilon_{R_t}$
- Background: $\mathbf{X}_0 = \mathbf{X}_B + \varepsilon_B$

Bayesian Inversion

- Likelihood and prior model: $p(\mathbf{Y}|\mathbf{X})$, $p(\mathbf{X})$
- Maximize posterior: $p(\mathbf{X}|\mathbf{Y})$ over $\mathbf{X}$
- Bayes rule: $\log p(\mathbf{X}|\mathbf{Y}) = \log p(\mathbf{Y}|\mathbf{X}) + \log p(\mathbf{X}) + \text{cste}$
- Variational inversion: $\nabla_{\mathbf{X}} \log p(\mathbf{X}|\mathbf{Y}) = \nabla_{\mathbf{X}} \log p(\mathbf{Y}|\mathbf{X}) + \nabla_{\mathbf{X}} \log p(\mathbf{X})$

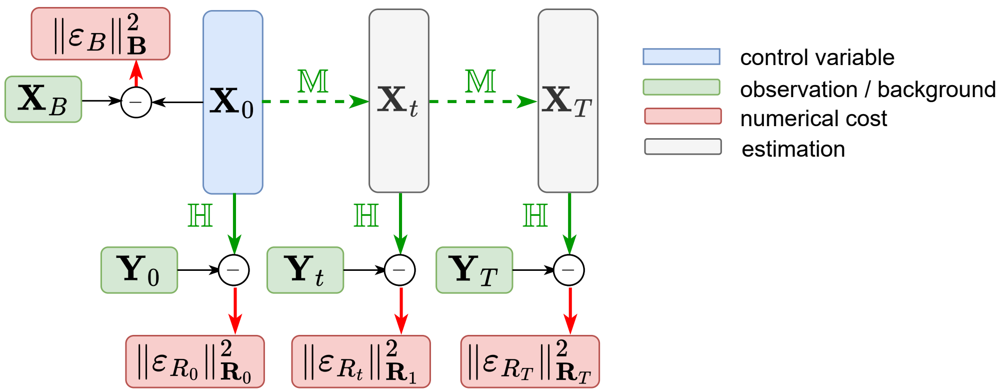
4DVAR - Maximum A Posteriori (MAP) estimation

- Gaussian error modeling: $\varepsilon_{R_t} \sim \mathcal{N}(0, \mathbf{R}_t)$, $\varepsilon_B \sim \mathcal{N}(0, \mathbf{B})$

$$-\log p(\mathbf{X} \mid \mathbf{Y}) = \underbrace{\frac{1}{2}\|\mathbf{X}_0 - \mathbf{X}_B\|^2_{\mathbf{B}}}_{\text{fit-to-prior}} + \underbrace{\frac{1}{2}\sum_{t=0}^{T}\|\mathbb{H}(\mathbf{X}_t) - \mathbf{Y}_t\|^2_{\mathbf{R}_t}}_{\text{fit-to-data}} \quad \text{s.t.} \quad \mathbf{X}_{t+1} = \mathbb{M}(\mathbf{X}_t)$$

4DVAR computational graph:

- strong constraint $p(\mathbf{X} \mid \mathbf{Y}) = p(\mathbf{X}_0 \mid \mathbf{Y})$
- optimal control problem



Deep Learning-like: adjoint state method $\approx$ backpropagation algorithm

Motivations

- **MAP** as a **point estimate** i) **can over-fit** ii) **does not quantify uncertainty**
- Can we design a **4DVAR-like** algorithm overcoming these issues ?

Variational Inference

Variational Inference:

- posterior distribution $p(\mathbf{X} \mid \mathbf{Y})$ is **intractable**
- Choice of **parameterized approximate** $q_\theta(\mathbf{X}) \approx p(\mathbf{X} \mid \mathbf{Y})$

Kullback–Leibler divergence:

- **statistical distance** between probability distribution
- $q_\theta^* = \arg\min_\theta D_{\mathcal{KL}}(q_\theta(\mathbf{X}) \,\|\, p(\mathbf{X} \mid \mathbf{Y}))$

$$D_{\mathcal{KL}}(q_\theta(\mathbf{X}) \,\|\, p(\mathbf{X} \mid \mathbf{Y})) = \underbrace{\mathbb{E}_{q_\theta}[\log q_\theta(\mathbf{X})] - \mathbb{E}_{q_\theta}[\log p(\mathbf{X}, \mathbf{Y})]}_{-ELBO} + \underbrace{\log p(\mathbf{Y})}_{\text{log-evidence}}$$

Evidence Lower Bound (ELBO):

- log-evidence $\log p(\mathbf{Y})$ is not computable but does not depend on θ
- minimizing $D_{\mathcal{KL}}(q_\theta(\mathbf{X}) \,\|\, p(\mathbf{X} \mid \mathbf{Y}))$ is equivalent to minimizing $-ELBO(\theta)$

$$-ELBO(\theta) = \underbrace{D_{\mathcal{KL}}(q_\theta(\mathbf{X}) \,\|\, p(\mathbf{X}))}_{\text{fit-to-prior}} - \underbrace{\mathbb{E}_{q_\theta}[\log p(\mathbf{X} \mid \mathbf{Y})]}_{\text{fit-to-data}}$$

Variational Inference: A Review for Statisticians [Blei et al, 2018]

Variational Inference 4DVAR:

▷ **strong constraint**: $q_\theta(\mathbf{X}) = q_\theta(\mathbf{X}_0)$

▷ **Gaussian modelling**: $-\log p(\mathbf{X} \mid \mathbf{Y}) = \frac{1}{2}\|\varepsilon_B\|^2_{\mathbf{B}} + \frac{1}{2}\sum_{t=0}^{T}\|\varepsilon_{R_t}\|^2_{\mathbf{R}_t} = \mathcal{L}_{4dv}(\mathbf{X}_0)$

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{X}_0\sim q_\theta}[\mathcal{L}_{4dv}(\mathbf{X}_0)] + \mathbb{E}_{\mathbf{X}_0\sim q_\theta}[\log q_\theta(\mathbf{X}_0)]$$

encourage q_θ to place its mass on **MAP** estimate

negative entropy: encourage q_θ to be diffuse

Sanity check:

▷ if $q_\theta(\mathbf{X}_0) = \delta(\theta - \mathbf{X}_0)$

▷ then $\mathbb{E}_{\mathbf{X}_0\sim q_\theta}[\log q_\theta(\mathbf{X}_0)] = 0$ and $\mathbb{E}_{\mathbf{X}_0\sim q_\theta}[\mathcal{L}_{4dv}(\mathbf{X}_0)] = \mathcal{L}_{4dv}(\mathbf{X}_0)$

▷ so $\mathcal{L}(\theta) = \mathcal{L}_{4dv}(\mathbf{X}_0)$

▷ **we recover 4DVAR loss function!**

Optimization? [Kingma & Weilling, 2013]

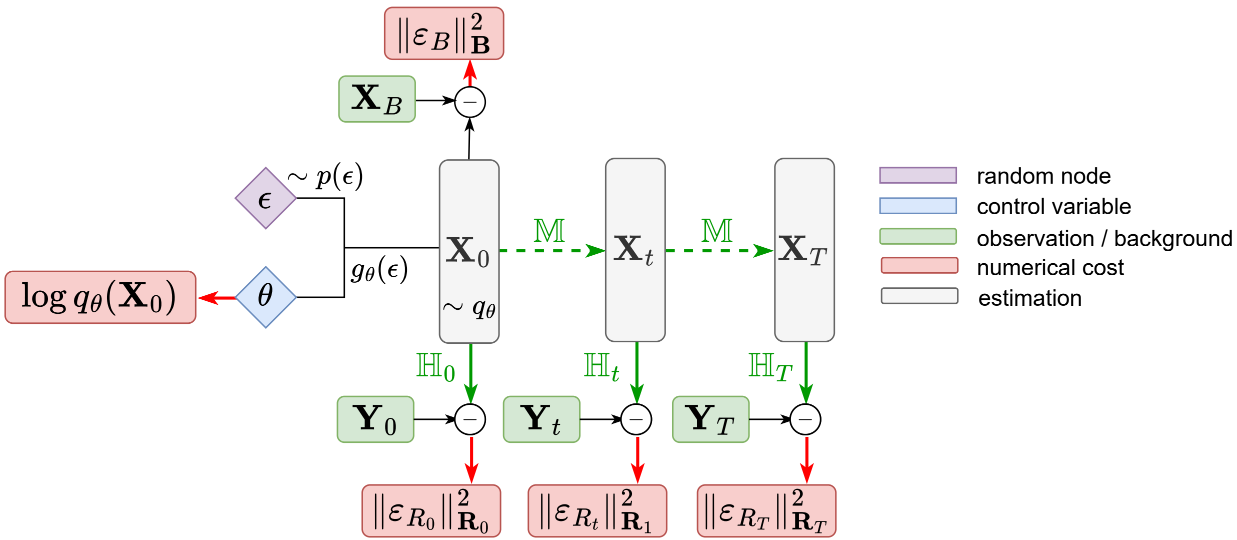
▷ **issue:** $\nabla_\theta \mathbb{E}_{q_\theta}[f_\theta] \neq \mathbb{E}_{q_\theta}[\nabla_\theta f_\theta]$

▷ **re-parametrization:** $\mathbf{X}_0 \sim q_\theta(\mathbf{X}_0)$ as $\mathbf{X}_0 = g_\theta(\epsilon)$ with $\epsilon \sim p(\epsilon)$ and g_θ **differentiable**

▷ **Monte Carlo estimate:** $\nabla_\theta \mathcal{L}(\theta) \approx \frac{1}{N} \sum_{\epsilon \sim p(\epsilon)} \left(\nabla_\theta \mathcal{L}_{4dv}\left(g_\theta(\epsilon^{(n)})\right) + \nabla_\theta \log q_\theta\left(g_\theta(\epsilon^{(n)})\right)\right)$

▷ **automatic differentiation** and **stochastic gradient descent**

VI-4DVAR computational graph: (N=1)



Stochastic / Black Box / Automatic differentiation Variational Inference [Blei et al]

II. Case study

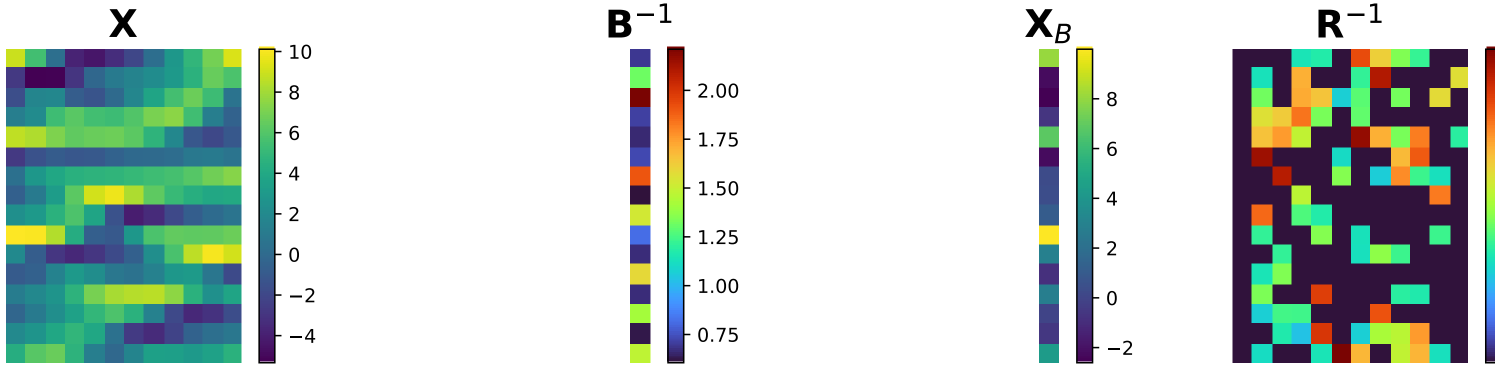
Case study - Twin experiment

Twin experiment:

- ▷ dynamical model: $\mathbb{M}$: Lorenz96 (RK4 scheme)
- ▷ observation operator: $\mathbb{H} =$ "**linear projection**" $\circ$ "**quadratic non-linearity**"
- ▷ Gaussian errors: $\varepsilon_{R_t} \sim \mathcal{N}(0, \mathbf{R}_t)$, $\varepsilon_B \sim \mathcal{N}(0, \mathbf{B})$

Example of simulated data:

- ▷ chaotic regime
- ▷ noises with different statistics at each grid point
- ▷ goal: estimate $p(\mathbf{X}_0 \mid \mathbf{Y}_{0:T})$



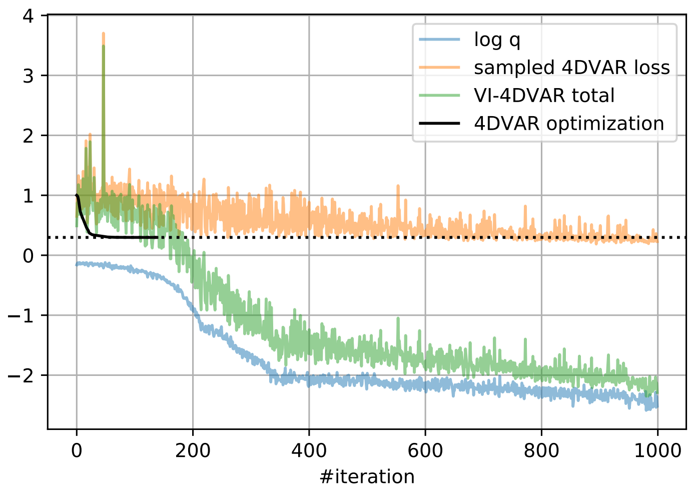
Variational Inference 4DVAR:

- ▷ Gaussian variational posterior: $q_\theta(\mathbf{X}_0) \sim \mathcal{N}(\mu, \Sigma)$
- ▷ Gaussian log-likelihood: $\log q_\theta(\mathbf{X}_0) = -\frac{1}{2}\|\mathbf{X}_0 - \mu\|^2_\Sigma - \frac{1}{2}\log|\Sigma| + cste$
- ▷ mean-field approximation: $q_\theta(\mathbf{X}_0) = \Pi q_{\theta_i}(\mathbf{x}_{0,i})$ *i.e. posterior covariance is diagonal*
- ▷ control parameters: $\theta = (\mu, diag(\Sigma) = \sigma^2)$

Re-parametrization "trick":

- ▷ $\epsilon \sim \mathcal{N}(0, 1)$ and $\mathbf{X}_0 = \mu + \epsilon \odot \sigma$ gives $\mathbf{X}_0 \sim \mathcal{N}(\mu, \Sigma)$

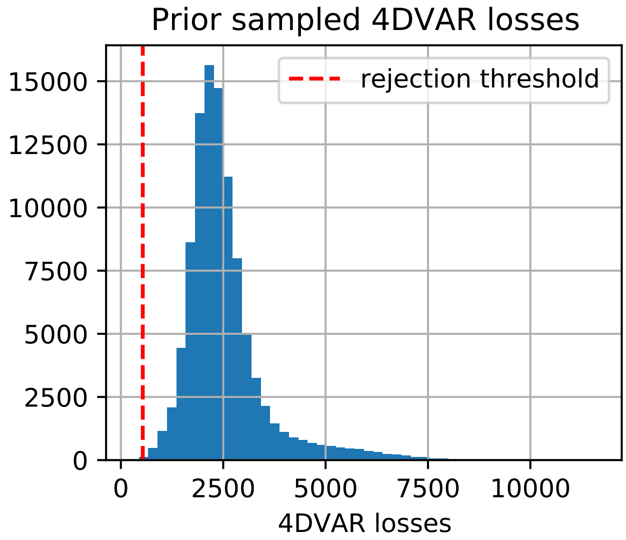
Optimization:

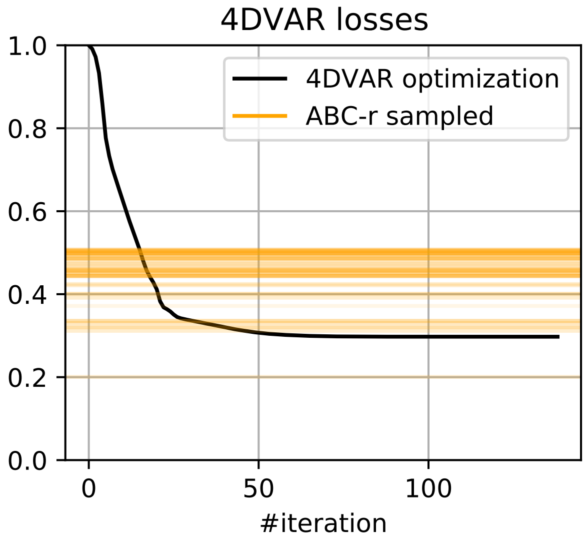


Limitation: costly in model forward

<u>**Approximate Bayesian Computation - Rejection sampling**</u>:

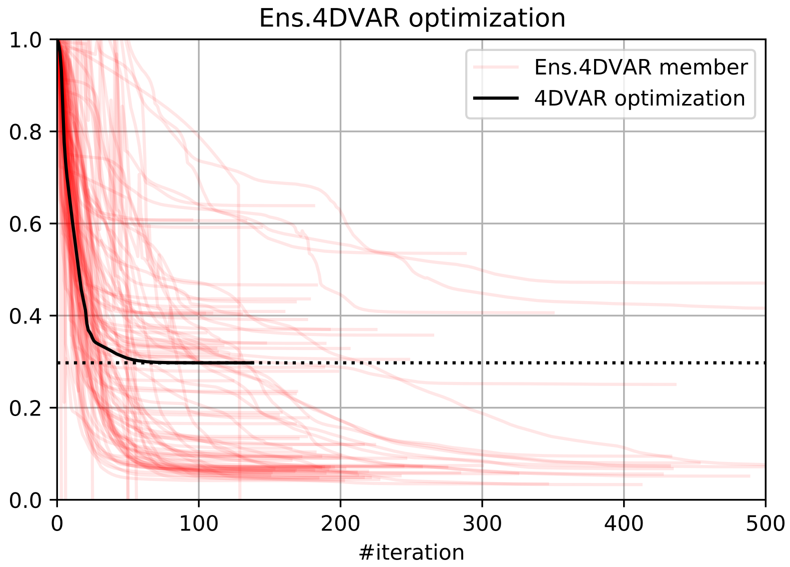
- ▷ **sample** from the background: $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{X}_B, \mathbf{B})$
- ▷ **compute** $\mathcal{L}_{4dv}(\mathbf{X}_0)$
- ▷ **reject** if $\mathcal{L}_{4dv}(\mathbf{X}_0) > \textit{threshold}$



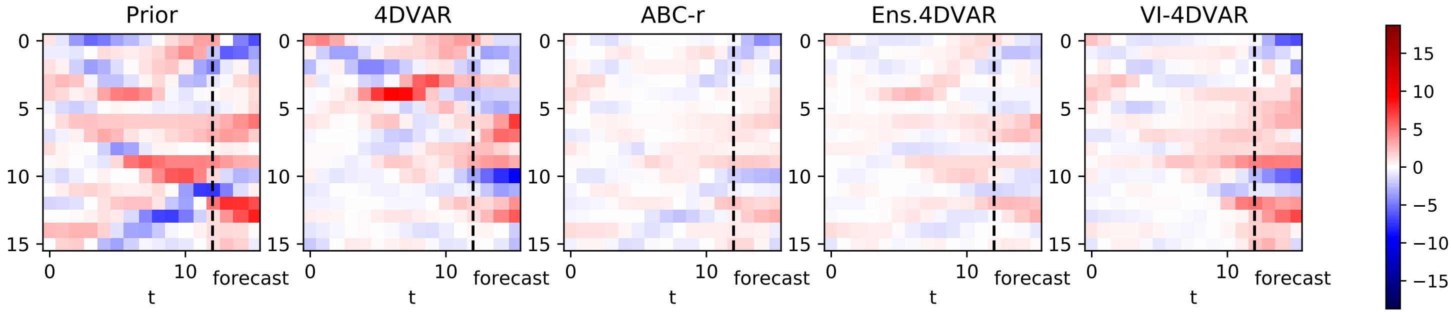


<u>**Ensemble of 4DVAR**</u>: [Jardak et Tallagrand, 2018]

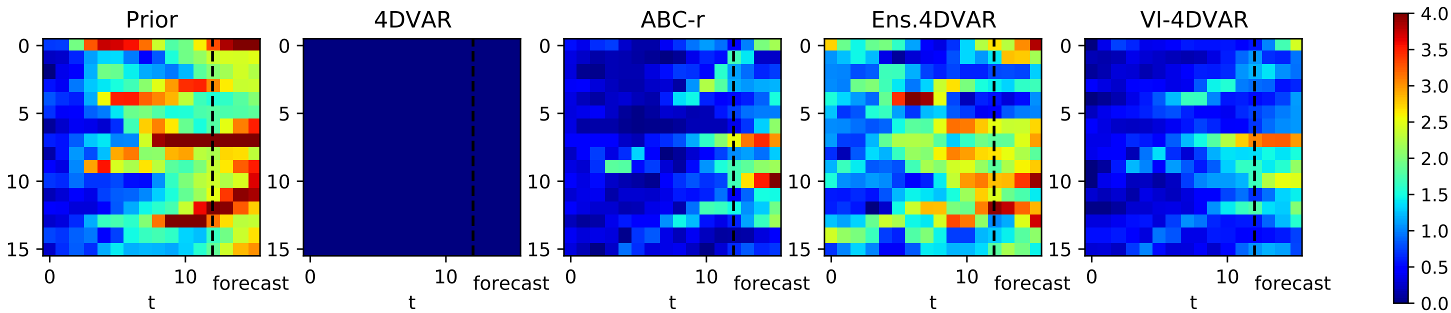
- ▷ **perturb** the background: $\mathbf{X}_B^{'} \sim \mathcal{N}(\mathbf{X}_B, \mathbf{B})$
- ▷ **perturb** the observation: $\mathbf{Y}^{'} \sim \mathcal{N}(\mathbf{Y}, \mathbf{R})$
- ▷ Optimize 4DVAR$(\mathbf{X}_B^{'}, \mathbf{Y}^{'})$



Error of the average:

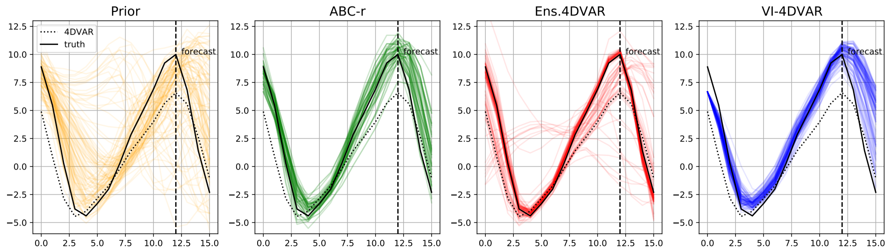


Standard deviation:

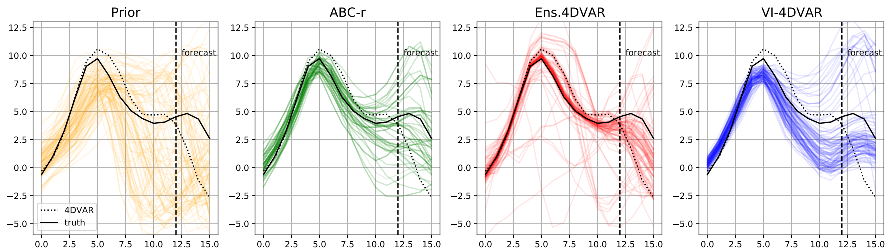


Sampled trajectory:

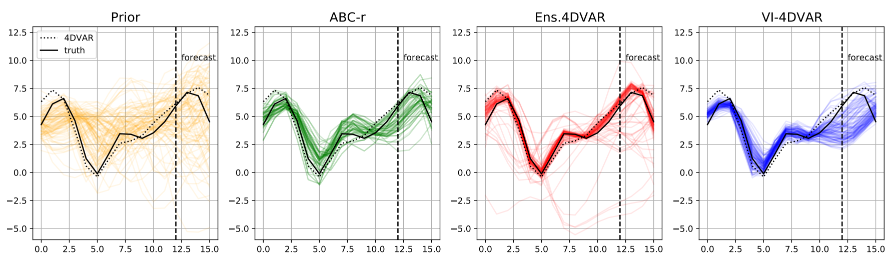
- ▷ coordinate 0



- ▷ coordinate 7

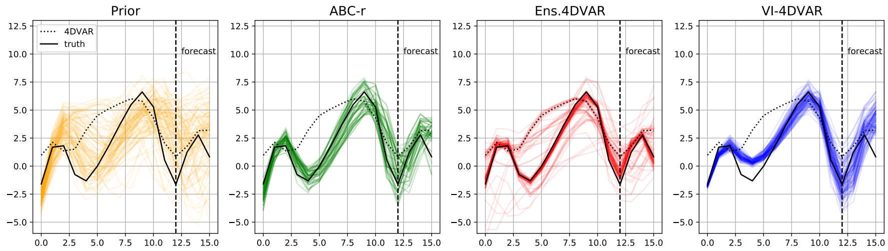


- ▷ coordinate 15 (last)

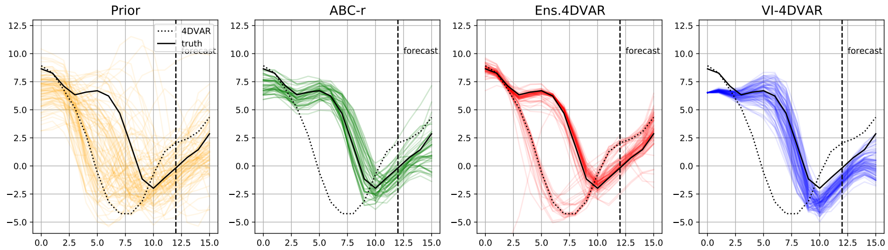


Sampled trajectory:

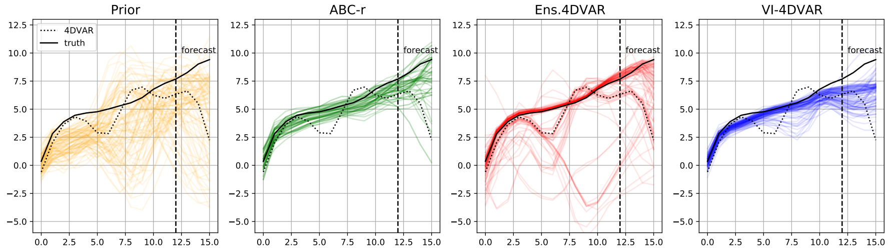
▷ coordinate 2



▷ coordinate 4



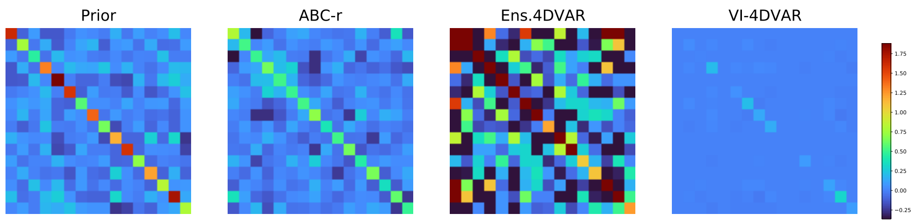
▷ coordinate 6



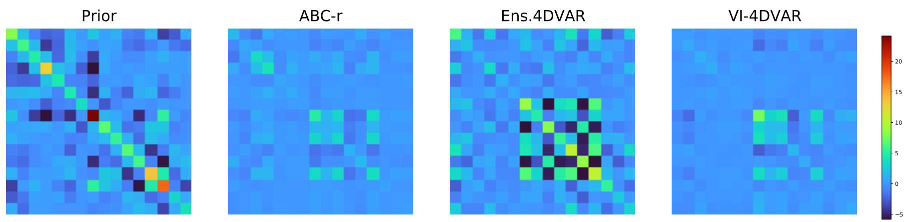
Sample covariance matrix:

- $t = 0$

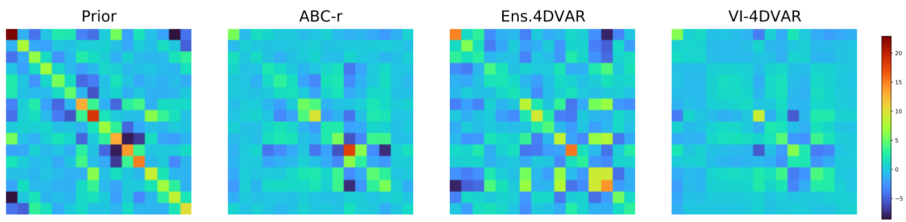
Limitation: only variance



- $t = 11$ (end of assimilation window)



- $t = 15$ (forecast)



III. Perspectives

Normalizing flows:

- ▷ flexible and arbitrarily complex approximate posterior distributions
- ▷ **simple** initial density is **transformed** into a more complex one
- ▷ applying **sequence of invertible transformation** (rule for change of variables)

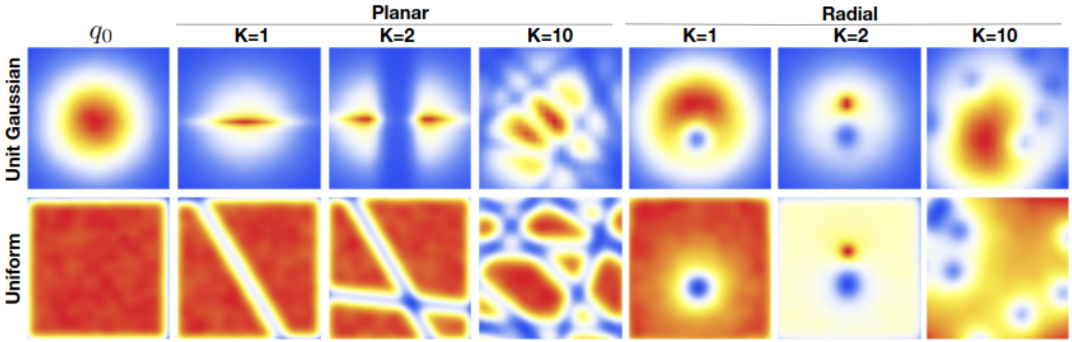


Figure 1. Effect of normalizing flow on two distributions.

from ***Variational Inference with Normalizing Flows*** [Rezende et Shakir, 2015]

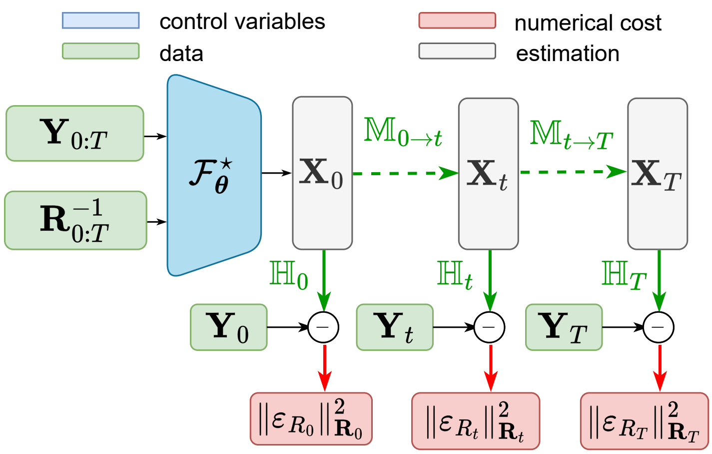
Perspectives

Amortized Inference:

- ▷ Motivation: optimizing a model on one data point is expensive
- ▷ Introduce a parametric family of conditional densities $q_\theta(\mathbf{X})$
- ▷ Learn a **recognition model** $g_\phi : \mathbf{Y} \mapsto \theta$

Learning 4DVAR inversion directly from observations [2023]

- ▷ Recognition network $\mathcal{F}^\star_{\boldsymbol{\theta}} : (\mathbf{Y}, \mathbf{R}^{-1}) \mapsto \mathbf{X}_0$
- ▷ Optimized on a **dataset**
- ▷ Variational posterior is a **delta** distribution



Perspectives

Accounting for model errors:

Weak constraint 4DVAR

- ▷ Dynamics: $\mathbf{X}_{t+1} = \mathbb{M}(\mathbf{X}_t) + \varepsilon_{m_t}$
- ▷ Gaussian error modeling: $\varepsilon_B \sim \mathcal{N}(0, \mathbf{B})$, $\varepsilon_{m_t} \sim \mathcal{N}(0, \mathbf{Q}_t)$, $\varepsilon_{R_t} \sim \mathcal{N}(0, \mathbf{R}_t)$

$$-\log p(\mathbf{X} \mid \mathbf{Y}) = \underbrace{\frac{1}{2}\|\varepsilon_B\|^2_{\mathbf{B}} + \frac{1}{2}\sum_{t=0}^{T-1}\|\varepsilon_{m_t}\|^2_{\mathbf{Q}_t}}_{\text{fit-to-prior}} + \underbrace{\frac{1}{2}\sum_{t=0}^{T}\|\varepsilon_{R_t}\|^2_{\mathbf{R}_t}}_{\text{fit-to-data}} \quad \text{s.t.} \quad \mathbf{X}_{t+1} = \mathbb{M}_t(\mathbf{X}_t) + \varepsilon_{m_t}$$

Variational Inference ?

- ▷ **proposed method naturally extends**, only the prior changes
- ▷ **simultaneous estimation of state and parameters**
- ▷ what model for the variational posterior ?

Take home message

You can optimize 4DVAR cost over the variational parameters of a distribution instead of the initial conditions

Code & Slides: `https://github.com/ArFiloche/VI-4DVAR`

$\longrightarrow$ *notebook_demo/ISDA_online.ipynb*

Thank you for your attention

`arthur.filoche@uwa.edu.au`

Experience parameters

```python
###### Parameters to play with #####

### Data ###

# Truth
Nx = 16 #state dim
Tw = 16 #time window
T = 12 #time assimilation (the rest is kept for forecast)

#Observation
p_drop = 0.5 #percentage drop in obs
subsample_t = 1 #subsampling factor in time (drop all columns)

sigma_perc_b = [10,20] #interval of percentage noise percentage in background
sigma_perc_obs = [5,10] #interval of percentage noise percentage in observations

def h_nonlin(x): # non-linearity in the observation operator

    return x**2

### Assimilation algorithm ###

# ABC-rejection sampling
N_trial_abc = 100000 #number of trials
percent_select_abc = 0.1 #percentage of candidate to select for the posterior distribution
percent_select_prior = 0.1 #percentage of candidate to select for the prior distribution

# Ensemble of 4DVAR
N_member_E4dv = 100

# VI-4DVAR
N_iter = 1500 #number of forward during the optimization
lr = 0.01 #learning rate of Adam optimizer
batch_size = 1 #number of sample for the Monte Carlo estimate of the gradient
N_vi_sample = 1000 #number of trajectory to sample after optimization
```