# ML-Assisted Resampling for Stochastic Parameterization with Memory

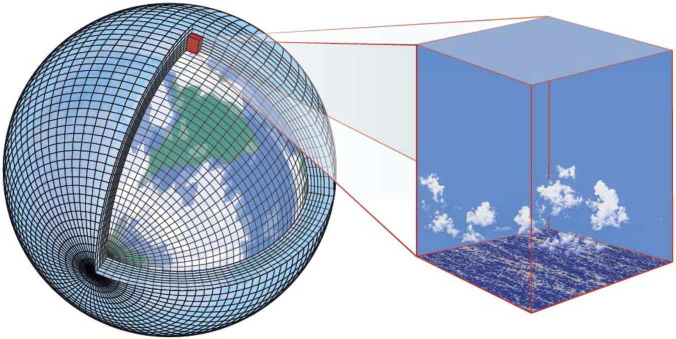

Daan Crommelin

Centrum Wiskunde & Informatica, Amsterdam
KdV Institute for Mathematics, University of Amsterdam

in collaboration with Wouter Edeling & Nick Verheul


CriticalEarth meeting, 25-29 april 2022

# Small-scale processes in atmosphere-ocean modeling



*(Figure: Schneider et al., 2017)*

# Simple example: 2-d non-divergent flow

PDE: $\boxed{\partial_t q + J(\psi, q) = f + \nu \Delta q}$ with $\Delta = \partial_{xx} + \partial_{yy}$

Streamfunction $\psi(x, y, t)$, vorticity $q(x, y, t)$, $q = \Delta\psi$

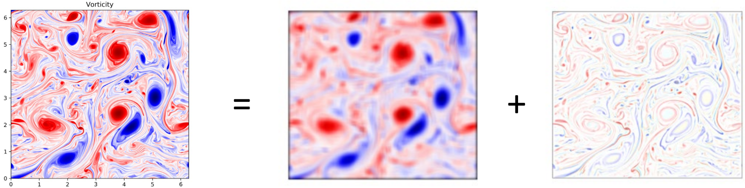
Velocity field: $(u, v) = (-\partial_y \psi,\ \partial_x \psi) \Rightarrow \nabla \cdot (u, v) = 0$

Forcing $f(x, y, t)$, viscosity $\nu$

**Nonlinear advection term**: $J(\psi, q) = (\partial_x \psi)(\partial_y q) - (\partial_y \psi)(\partial_x q) = (u, v) \cdot \nabla q$

# Example: 2-d non-divergent flow

Decompose: $q = \bar{q} + q'$ (large-scale + small-scale)




$$R(x, y, t) := J(\bar{\psi}, \bar{q}) - \overline{J(\psi, q)} \neq 0 \quad \rightarrow \quad \boxed{\partial_t \bar{q} + J(\bar{\psi}, \bar{q}) = \bar{f} + \nu \Delta \bar{q} + R}$$

**unclosed equation**

# Parameterization

Parameterization: representing unresolved processes

- **Stochastic.** e.g. Palmer, 2001; Berner et al., 2017

STOCHASTIC PARAMETERIZATION
Toward a New View of Weather and Climate Models

Judith Berner, Ulrich Achatz, Lauriane Batté, Lisa Bengtsson, Alvaro de la Cámara,
Hannah M. Christensen, Matteo Colangeli, Danielle R. B. Coleman, Daan Crommelin,
Stamen I. Dolaptchiev, Christian L. E. Franzke, Petra Friederichs, Peter Imkeller, Heikki Järvinen

- **Data-driven.**
  - Fitting stochastic processes: e.g. Wilks, 2005; Crommelin and Vanden-Eijnden, 2008
  - Deep learning: e.g. Rasp et al., 2018; Bolton and Zanna, 2019

PNAS

**Deep learning to represent subgrid processes in climate models**

Stephan Rasp[a,b,1], Michael S. Pritchard[b], and Pierre Gentine[c,d]

[a]Meteorological Institute, Ludwig-Maximilian-University, 80333 Munich, Germany; [b]Department of Earth System Science, University of California, Irvine, CA 92697; [c]Department of Earth and Environmental Engineering, Earth Institute, Columbia University, New York, NY 10027; and [d]Data Science Institute, Columbia University, New York, NY 10027

Edited by Isaac M. Held, Geophysical Fluid Dynamics Laboratory, National Oceanic and Atmospheric Administration, Princeton, NJ, and approved August 8, 2018 (received for review June 14, 2018)

# Multiscale dynamical system

Model set-up, in ODE form (e.g. from discretizing PDE):

A multiscale system with macroscopic degrees of freedom $x$ and micropscopic degrees of freedom $y$

$$\begin{aligned} \tfrac{d}{dt}x &= f(x,\sigma) \\ \tfrac{d}{dt}y &= g(x,y) \\ \sigma &= \sigma(y) \end{aligned}$$

Two-way coupling, $\sigma(y)$ represents micro-to-macro coupling/feedback

# Parameterization

Focus here on **additive** feedback, $f(x, \sigma)$ becomes $f(x) + \sigma$

Discrete-time version (constant time step), time index $j \in \mathbb{N}$ :

$$\begin{aligned} x_{j+1} &= F(x_j) + r_j \\ y_{j+1} &= G(x_j, y_j) \\ r_j &:= r(y_j) \end{aligned}$$

**Aim**: long simulations of $x$ without resolving $y$

$\rightarrow$ parameterize feedback/forcing $r$ in terms of $x$

# Stochastic parameterization with memory

Evolve / update $r$ by random sampling from conditional distribution

$r_{j+1} \mid r_j, r_{j-1}, ..., x_j, x_{j-1}, ...$

in tandem with $x$ updates according to $x_{j+1} = F(x_j) + r_j$

$\rightarrow$ **Stochastic** parameterization of $r$, with **memory** (cf. Mori-Zwanzig theory)

In general, distribution $r_{j+1} \mid r_j, r_{j-1}, ..., x_j, x_{j-1}, ...$ is unknown.
However, we only need to *sample* from it.

# Data-driven approach

Assume we have **observations** of $x$ and $r$:

$(x_j^o, r_j^o)$, $j = 0, 1, ..., T$

Data source:
e.g. high-resolution simulations (with $y$ resolved) limited in space/time,
or $x$-only measurements (using $r_j^o = x_{j+1}^o - F(x_j^o)$ to get $r$ observations)

Use **resampling** of observations to approximate sampling from $r_{j+1} \mid r_j, r_{j-1}, ..., x_j, x_{j-1}, ...$

# Resampling

Reduced model with resampling parameterization:

$$\tilde{x}_{j+1} = F(\tilde{x}_j) + \tilde{r}_j \quad \tilde{r}_{j+1} : \text{random sample from set } \left\{ \forall\ r^o_{k+1} \;\middle|\; \tilde{d}_j \text{ close to } d^o_k \right\}$$

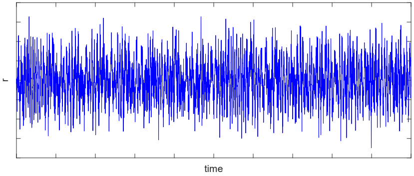
with **feature vectors**

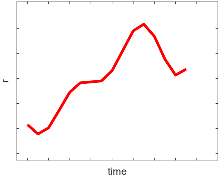
$$\tilde{d}_j := (\tilde{r}_j, \tilde{r}_{j-1}, \ldots, \tilde{r}_{j-J}, \tilde{x}_j, \tilde{x}_{j-1}, \ldots, \tilde{x}_{j-J})$$
$$d^o_j := (r^o_j, r^o_{j-1}, \ldots, r^o_{j-J}, x^o_j, x^o_{j-1}, \ldots, x^o_{j-J})$$

and memory depth $J$

(cf. local Markov bootstrap (Paparoditis & Politis, 2002), kNN resampling (Lall & Sharma, 1996))

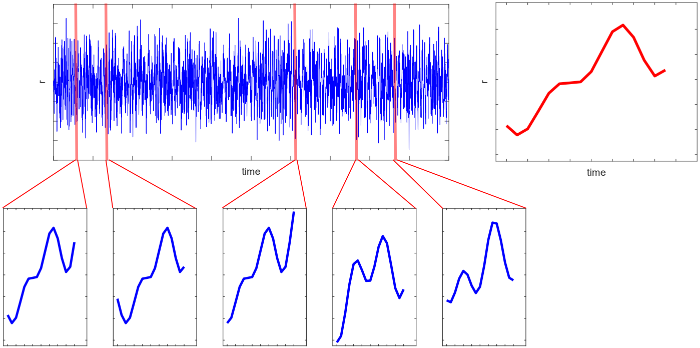
# Resampling: example



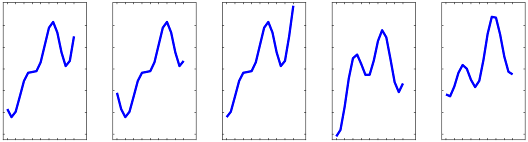



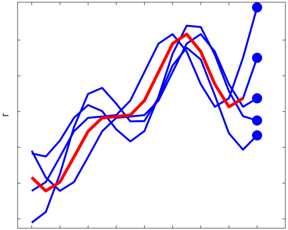

Blue: observation timeseries ($r_k^o$)
Red: reduced model recent history ($\tilde{r}_j$)

# Resampling: example

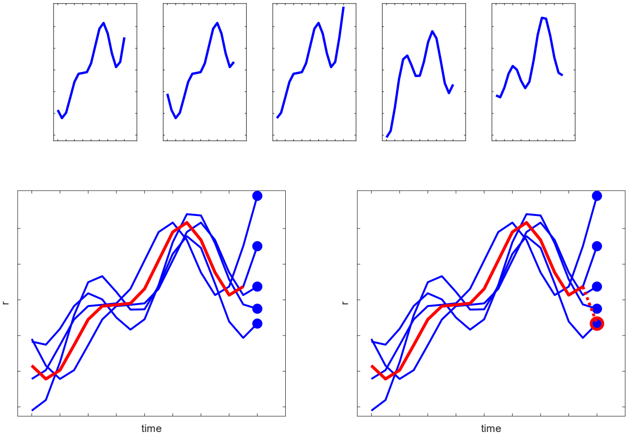


# Resampling: example





# Resampling: example



# Resampling by binning

Let $D$ denote the feature space, i.e. $d_j^o \in D$ and $\tilde{d}_j \in D$

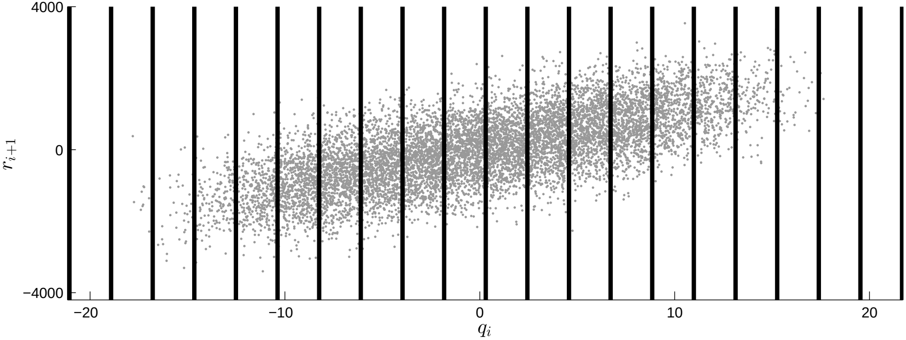
Discretize $D$ with $M$ bins: $D = D_1 \cup D_2 \cup ... \cup D_M$

Define $R_m := \{\forall\ r_{j+1}^o \,|\, d_j^o \in D_m\}$

Resampling:
1. find $n$ such that $\tilde{d}_j \in D_n$
2. sample $\tilde{r}_{j+1}$ randomly from $R_n$

# Resampling by binning

Example with dim($D$)=1:




*(Verheul & Crommelin, Comm Math Sci, 2016)*

Good results with Kac-Zwanzig heat bath model, with dim($D$)=3
but: curse of dimension if dim($D$) $\gg$ 1

# Binning the output

**Alternative** to binning space of $d_j^o, \tilde{d}_j$: binning space of $r_{j+1}^o, \tilde{r}_{j+1}$

$r_{j+1}^o \in B_1 \cup B_2 \cup ... \cup B_M$ for all $j$

Train neural net to map $d_j^o$ to discrete probability distribibution $\rho = (\rho_1, ..., \rho_M)$ over the bins:

$$\rho_m^{\mathrm{NN}}(d_j^o) \approx \mathbb{P}(r_{j+1}^o \in B_m \,|\, d_j^o)$$

Probabilistic classification with quantized softmax network (QSN)

(Crommelin & Edeling, Physica D, 2021)

# Resampling with QSN

In the reduced model, to generate $\tilde{r}_{j+1}$ given $\tilde{d}_j$:

(i) compute $\rho = \rho^{\text{NN}}(\tilde{d}_j)$
(ii) random sampling of $m \in \{1, 2, ..., M\}$ according to distribution $\rho$
(iii) random sampling of $\tilde{r}_{j+1}$ from all $r_k^o$ in bin $B_m$

**NB**: fundamentally different from direct NN prediction of $\tilde{r}_{j+1}$ from $\tilde{d}_j$

# Quantized Softmax Network

QSN:

input $d_j^o \in \mathbb{R}^{2NJ+2N}$ (feature vector)

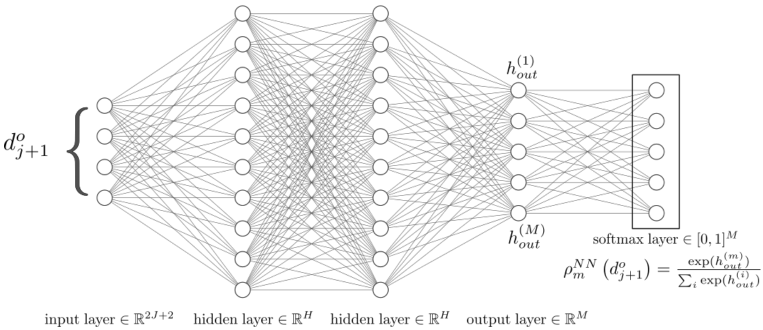
output $\rho \in [0,1]^M$ (probability mass function, $\sum_{m=1}^{M} \rho_m = 1$)

Feed-forward architecture with softmax-layer

Training with cross-entropy loss function

# Quantized Softmax Network

Architecture for $N = \dim(x_j) = \dim(r_j) = 1$:

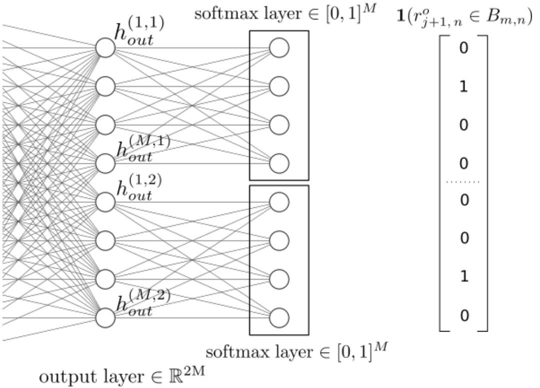



*(Figures: Crommelin & Edeling, Phys D, 2021)*

# QSN with $N > 1$

If $N > 1$, linear (in $N$) scaling of QSN input and output, by using different distributions for different elements of $r_j$

e.g. $N = 2$:



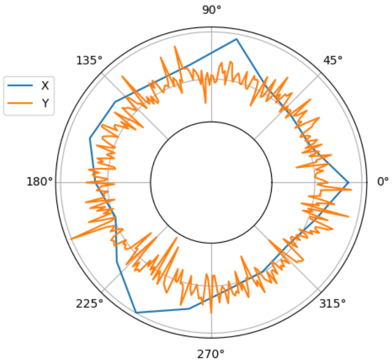
# Test case: L96 model

Two-layer Lorenz 96 model: coupled nonlinear ODEs (E.N. Lorenz, 1996)

$$\frac{d}{dt}X_k = X_{k-1}(X_{k+1} - X_{k-2}) - X_k + F + r_k$$

$$\frac{d}{dt}Y_{j,k} = \frac{1}{\varepsilon}\big(Y_{j+1,k}(Y_{j-1,k} - Y_{j+2,k}) - Y_{j,k} + h_y X_k\big)$$

with $r_k = \frac{h_x}{J}\sum_{j=1}^{J} Y_{j,k}$



# Test case: L96 model

Two-layer Lorenz 96 model: coupled nonlinear ODEs (E.N. Lorenz, 1996)

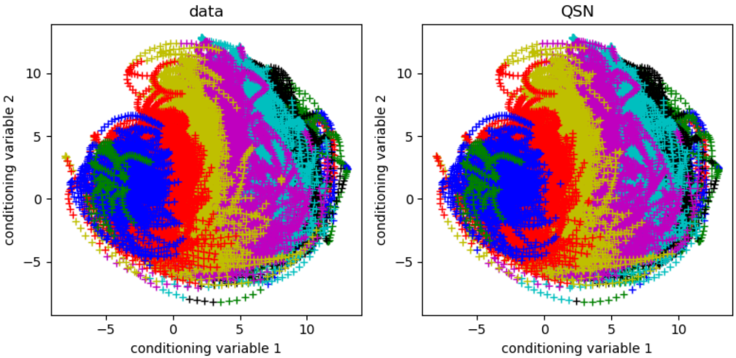
Two parameter settings considered:

(i) “Unimodal” setting: unimodal pdf for model variables, often used

(ii) “Bimodal” setting: non-standard, more challenging

$\varepsilon = 0.5$, $\dim(x) = \dim(r) = N = 18$ (and $\dim(y) = 360$)

# Test case: L96 model

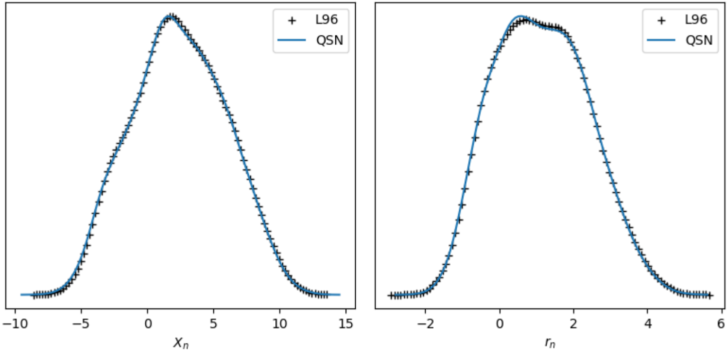



Left: bin index of training data. Right: QSN most probable bin

(x,y axis: elements of feature vector. Color: bin index)

# Test case: L96 model

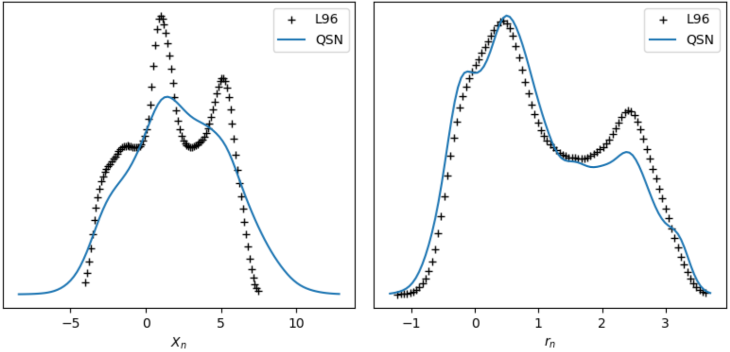
Full L96 model (+) and reduced model with QSN resampling parameterization (–)




PDFs of $x$ and $r$ from long time integration, unimodal setting

Feature vector: $d_j^o = (X(j), X(j-9))$

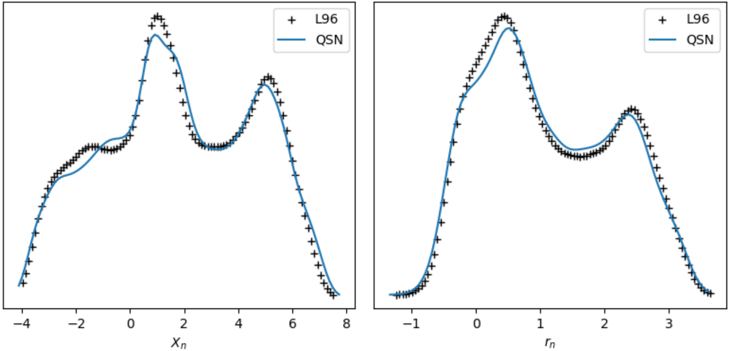
# Test case: L96 model




PDFs of $x$ and $r$, bimodal setting

Feature vector: $d_j^o = (X(j), X(j-1), ..., X(j-9))$

# Test case: L96 model




PDFs of $x$ and $r$, bimodal setting

Longer memory: feature vector $d_j^o = (X(j), X(j-1), ..., X(j-74))$

# Concluding remarks

- Machine learning for *stochastic* parameterization of small scales.
  Random resampling using QSN for probabilistic classification.
  (alternative approach using GANs: e.g. Gagne et al., 2020)

- Memory dependence through feature vector

- Physical consistency: only $r_j^o$ used, as observed from high-resolution model or physical measurements

# References and acknowledgements


- N. Verheul and D. Crommelin. *Data-driven stochastic representations of unresolved features in multiscale models*. Communications in Mathematical Sciences, 14(5), 2016.
- D. Crommelin and W. Edeling. *Resampling with neural networks for stochastic parameterization in multiscale systems*. Physica D, vol. 422, 132894, 2021

  (code on github.com/wedeling/EasySurrogate/tree/phys_D )


---


Funding acknowledgements:

The Netherlands Organization for Scientific Research (NWO),
Vidi project *Stochastic models for unresolved scales in geophysical flows*

The European Union's Horizon 2020 research and innovation programme,
grant agreement #800925 (VECMA project).