Very Large Parameter Problems

INI Event: The Role of Uncertainty in Mathematical Modelling of Pandemics

Wouter Edeling
CWI Amsterdam

February 10, 2022

This research received funding from the European Union Horizon 2020 research and innovation programme 800925 (VECMA project).

Recap dimension-adaptive SC

- Tuesday's talk: dimension-adaptive Stochastic Collocation (SC)

  Recap:

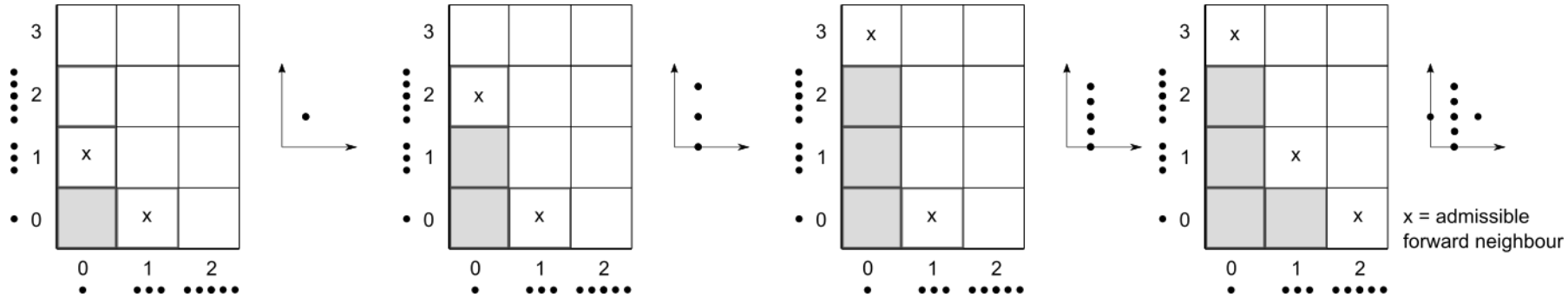
- SC: relies on 1 dimensional quadrature points
- $1 \rightarrow D$ dimensions: tensor products of 1D points (expensive!)

Recap dimension-adaptive SC

- Tuesday's talk: dimension-adaptive Stochastic Collocation (SC)

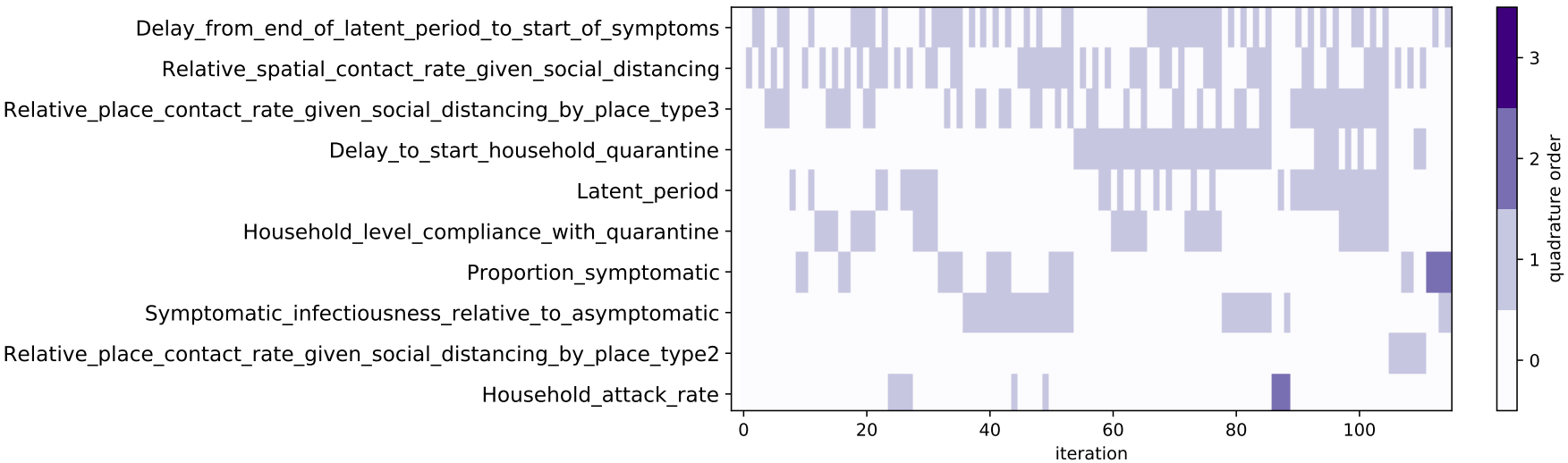
  Recap:

- SC: relies on 1 dimensional quadrature points
- $1 \rightarrow D$ dimensions: tensor products of 1D points (expensive!)
- Dimension-adaptive: iterative refinement sampling plan
- Only refines important inputs (based on error metric)



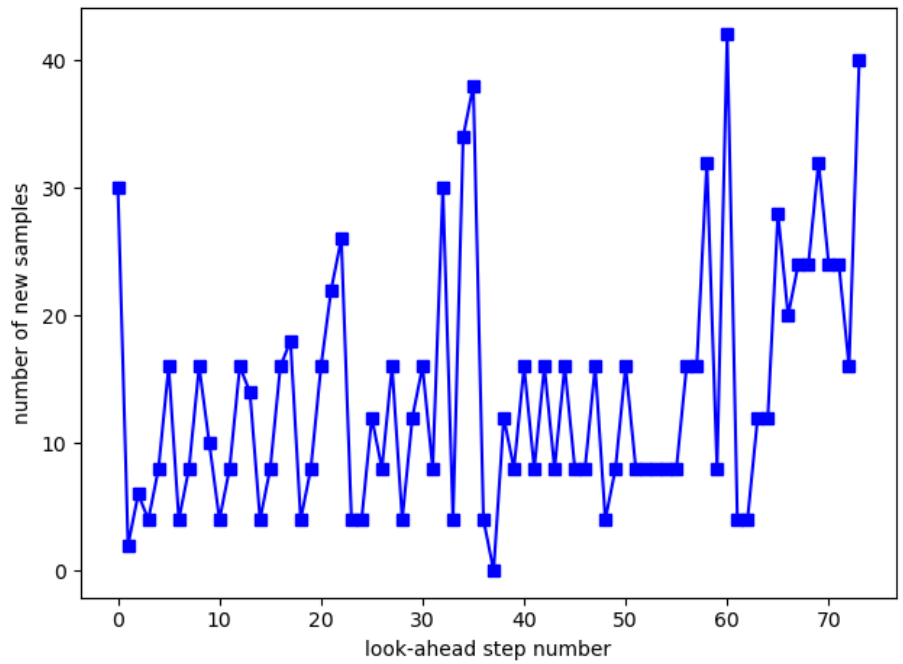
Limitations dimension-adaptive SC

- Still based on tensor products
- Iterative refinement also has a (potential) downside:



Limitations dimension-adaptive SC

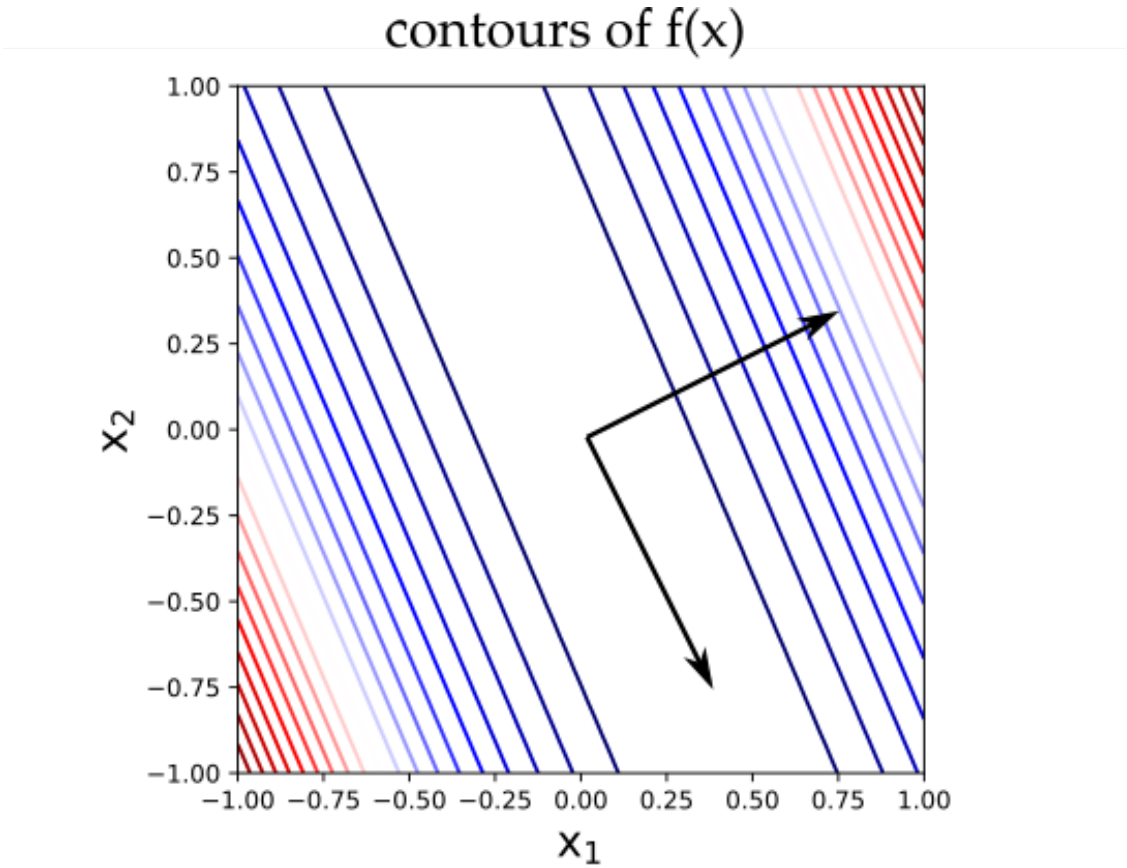
- Still based on tensor products
- Iterative refinement also has a (potential) downside:



- Small ensembles = many iterations.
- $\approx$ 10 new samples on average here (not very 'HPC friendly').
- On supercomputers queue time could become an issue.

Limitations dimension-adaptive SC

- In adaptive SC: coordinate axes are aligned with input x axes.
- Likely: direction of highest variability of $f(\mathrm{x})$ is not aligned



- Active subspace method: is there a $d < D$ dimensional subspace where $f(\mathrm{x})$ varies the most (on average)?

Active subspaces

Active subspace method: [1]

- Let $\mathrm{x} \in \mathbb{R}^D$, try to find active variables $\mathrm{y} \in \mathbb{R}^{\mathrm{d}}$, $d < D$:

$$\boxed{\mathrm{y} = U_1^T \mathrm{x}}, \quad U_1 \in \mathbb{R}^{D \times d}$$

[1] Constantine, P. G., Dow, E., & Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4)

Active subspaces

Active subspace method: [1]

- Let $\mathrm{x} \in \mathbb{R}^D$, try to find active variables $\mathrm{y} \in \mathbb{R}^{\mathrm{d}}$, $d < D$:

$$\boxed{\mathrm{y} = U_1^T \mathrm{x}}, \quad U_1 \in \mathbb{R}^{D \times d}$$

- Goal: approximate $f(\mathrm{x})$ in reduced input domain:

$$\boxed{f(\mathrm{x}) \approx G(\mathrm{y})}$$

[1] Constantine, P. G., Dow, E., & Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4)

Active subspaces

Active subspace method: [1]

- Let $\mathrm{x} \in \mathbb{R}^D$, try to find active variables $\mathrm{y} \in \mathbb{R}^{\mathrm{d}}$, $d < D$:

$$\boxed{\mathrm{y} = U_1^T \mathrm{x}}, \quad U_1 \in \mathbb{R}^{D \times d}$$

- Goal: approximate $f(\mathrm{x})$ in reduced input domain:

$$\boxed{f(\mathrm{x}) \approx G(\mathrm{y})}$$

Questions:

- What is U_1?
- What is $G(\cdot)$?

[1] Constantine, P. G., Dow, E., & Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4)

Active subspaces: finding U_1

- To find directions of greatest variability, consider:

$$C = \int (\nabla f(\mathrm{x}))(\nabla f(\mathrm{x}))^T \, p(\mathrm{x})\mathrm{dx}.$$

Active subspaces: finding U_1

ECMA

- To find directions of greatest variability, consider:

$$C = \int (\nabla f(\mathrm{x}))(\nabla f(\mathrm{x}))^T p(\mathrm{x})\mathrm{dx}.$$

- $p(\mathrm{x})$ is the input probability density function.
- C is a covariance-like matrix of the gradient $\nabla f(x)$

  $\rightarrow$ it has the following decomposition:

$$C = [U_1\ U_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_1\ U_2]^T$$

  $\rightarrow$ Λ_1 are the d largest eigenvalues.

  $\rightarrow$ U_1 are corresponding (orthonomal) eigenvectors.

  $\rightarrow$ U_1 points in direction of greatest (on-average) variability

Active subspaces: finding U_1

ECMA

- To find directions of greatest variability, consider:

$$C = \int (\nabla f(\mathrm{x}))(\nabla f(\mathrm{x}))^T p(\mathrm{x})\mathrm{dx}.$$

- $p(\mathrm{x})$ is the input probability density function.
- C is a covariance-like matrix of the gradient $\nabla f(x)$

  $\rightarrow$ it has the following decomposition:

$$C = [U_1\ U_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_1\ U_2]^T$$

  $\rightarrow$ Λ_1 are the d largest eigenvalues.

  $\rightarrow$ U_1 are corresponding (orthonomal) eigenvectors.

  $\rightarrow$ U_1 points in direction of greatest (on-average) variability

Active subspaces: finding G

- The active variables y are now defined:

$$C = [U_1 \; U_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_1 \; U_2]^T \quad \text{and} \quad \mathrm{y} = U_1^T \mathrm{x}$$

Active subspaces: finding G

- The active variables y are now defined:

$$C = [U_1\ U_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_1\ U_2]^T \quad \text{and} \quad \mathrm{y} = U_1^T \mathrm{x}$$

- What is $G(\mathrm{y}) \approx f(\mathrm{x})$?
  $\rightarrow$ any (surrogate) modelling method trained in y domain
  e.g. a Gaussian Process.

Active subspaces: finding G

ECMA

- The active variables y are now defined:

$$C = [U_1\ U_2] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [U_1\ U_2]^T \quad \text{and} \quad \mathrm{y} = U_1^T \mathrm{x}$$

- What is $G(\mathrm{y}) \approx f(\mathrm{x})$?
  $\rightarrow$ any (surrogate) modelling method trained in y domain
  e.g. a Gaussian Process.
- Potential problem:

$$C = \int (\nabla f(\mathrm{x}))\,(\nabla f(\mathrm{x}))^T\, p(\mathrm{x})\mathrm{dx}.$$

  Computing C requires the gradient of (computer code) $f(\mathrm{x})$.

Active subspaces: gradient requirement

- ∇f might not always be available / easy to compute.
- Alternatives without need for gradient information exist:
  - based on Gaussian Processes [2]
  - based on neural networks: deep-active subspaces (DAS) [3]
- We will focus on DAS.

[2] Liu, X., & Guillas, S. (2017). Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights. SIAM/ASA Journal on Uncertainty Quantification, 5(1), 787-812.

[3] Tripathy, R., & Bilionis, I. (2019, August). Deep active subspaces: A scalable method for high-dimensional uncertainty propagation. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Deep active subspaces

Similarities to active subspaces:

- Active variables are defined as $\boxed{\mathrm{y} = W_1^T \mathrm{x}}$,
- $W_1 \in \mathbb{R}^{D \times d}$, like U_1, has orthonormal columns,
- A surrogate is constructed as $\widetilde{G}(\mathrm{y}) \approx f(\mathrm{x})$

Deep active subspaces

Similarities to active subspaces:

- Active variables are defined as $\boxed{\mathrm{y} = W_1^T \mathrm{x}}$,
- $W_1 \in \mathbb{R}^{D \times d}$, like U_1, has orthonormal columns,
- A surrogate is constructed as $\widetilde{G}(\mathrm{y}) \approx f(\mathrm{x})$

Differences:

- $W_1 = W_1(Q)$ are Gram-Schmidt vectors, not eigenvectors,

$$\mathrm{w}_i = \mathrm{q}_i - \sum_{j=1}^{i-1} \left(\frac{\mathrm{w}_j^T \mathrm{q}_i}{\mathrm{w}_j^T \mathrm{w}_j} \right) \mathrm{w}_j, \quad i = 1, \cdots, d.$$

Deep active subspaces

Similarities to active subspaces:

- Active variables are defined as $\boxed{\mathrm{y} = W_1^T \mathrm{x}}$,
- $W_1 \in \mathbb{R}^{D \times d}$, like U_1, has orthonormal columns,
- A surrogate is constructed as $\widetilde{G}(\mathrm{y}) \approx f(\mathrm{x})$

Differences:

- $W_1 = W_1(Q)$ are Gram-Schmidt vectors, not eigenvectors,

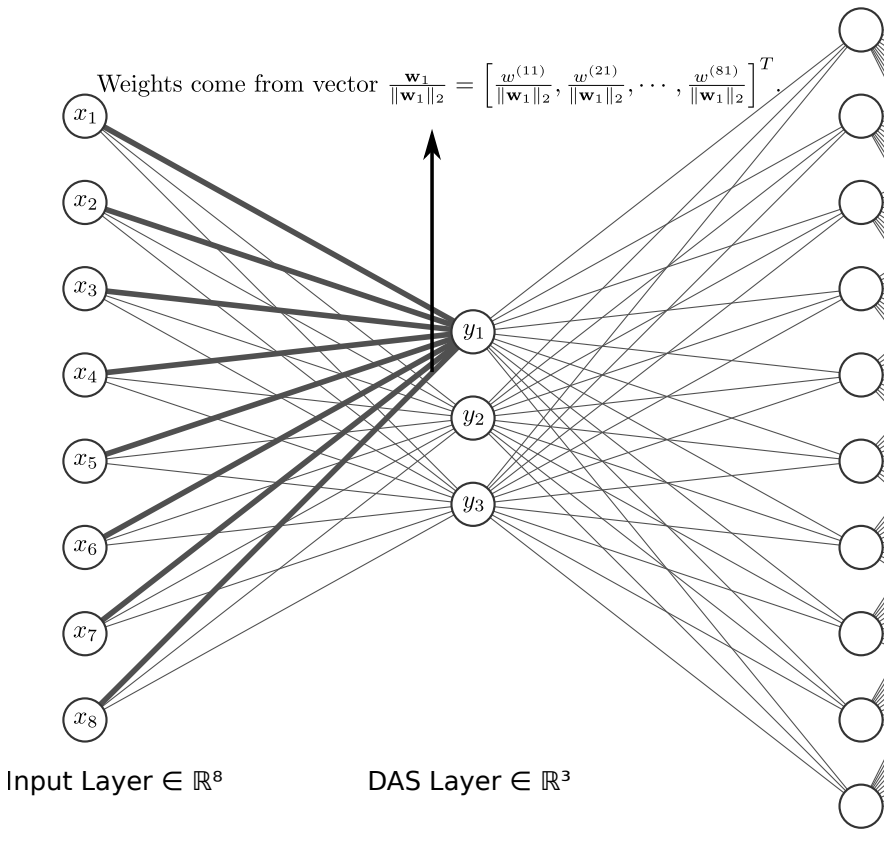
$$\mathrm{w}_i = \mathrm{q}_i - \sum_{j=1}^{i-1} \left(\frac{\mathrm{w}_j^T \mathrm{q}_i}{\mathrm{w}_j^T \mathrm{w}_j} \right) \mathrm{w}_j, \quad i = 1, \cdots, d.$$

- $W_1 =$ weights of neural network layer with linear activation Φ:

$$\mathrm{y} = \Phi\left(W_1^T \mathrm{x}\right) = W_1^T \mathrm{x}$$

Deep active subspaces: training

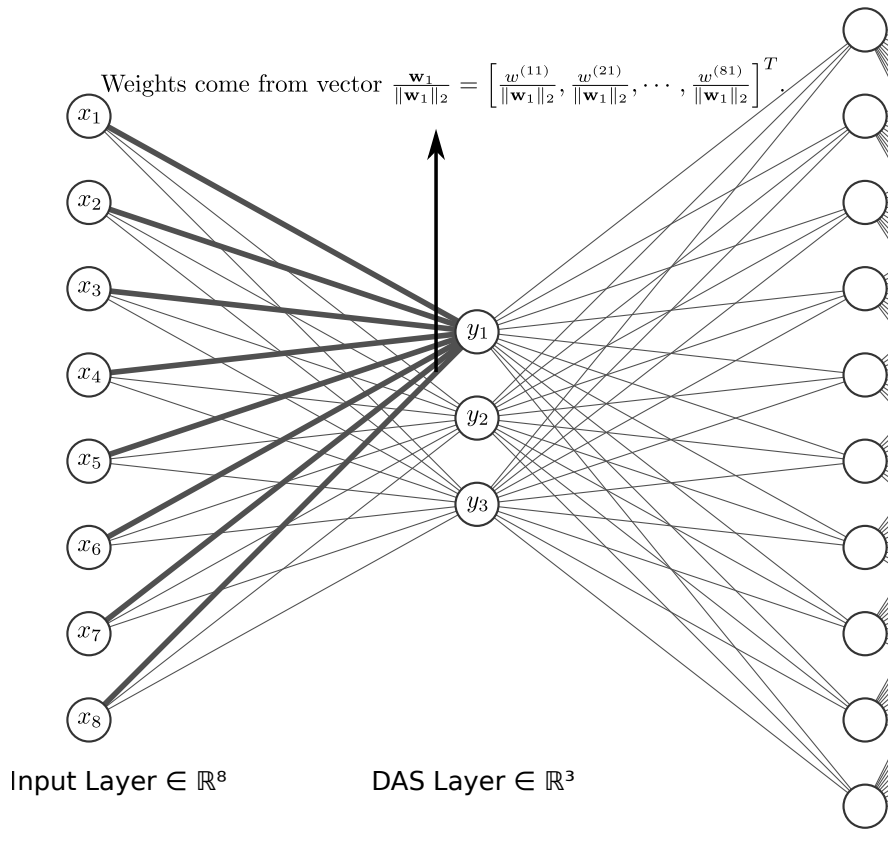
- We'll optimize $W_1(Q)$ using back propagation



- Squared loss:
$$L = \frac{1}{N}\sum_{i=1}^{N}\left(f(\mathrm{x}_i) - \widetilde{G}(\mathrm{y}_i)\right)^2$$
- Normal weight update:
$$W_i \leftarrow W_i - \alpha \partial L / \partial W_i$$

Deep active subspaces: training

- We'll optimize $W_1(Q)$ using back propagation



- Squared loss:
$$L = \frac{1}{N}\sum_{i=1}^{N}\left(f(\mathrm{x}_i) - \widetilde{G}(\mathrm{y}_i)\right)^2$$
- Normal weight update:
$$W_i \leftarrow W_i - \alpha \partial L/\partial W_i$$
- In DAS layer:
$$Q \leftarrow Q - \alpha \partial L/\partial Q$$
- Chain rule on (i,j)-th entry of $\partial L/\partial Q$: $\frac{\partial L}{\partial q_{ij}} = \frac{\partial L}{\partial W_1}\frac{\partial W_1}{\partial q_{ij}}$

Deep active subspaces: GS derivatives

$$\frac{\partial L}{\partial q_{ij}} = \frac{\partial L}{\partial W_1} \frac{\partial W_1}{\partial q_{ij}}$$

- Need the derivatives of Gram-Schmidt vectors: $\frac{\partial \mathrm{w}_i}{\partial \mathrm{q}_j}$
- $\mathrm{w}_i\left(\mathrm{q}_1, \mathrm{q}_2, \cdots, \mathrm{q}_i\right)$ becomes complicated expression for $i > 1$.
- Original DAS article used automatic differentiation

[4] https://github.com/wedeling/Gram_Schmidt_Derivatives

Deep active subspaces: GS derivatives

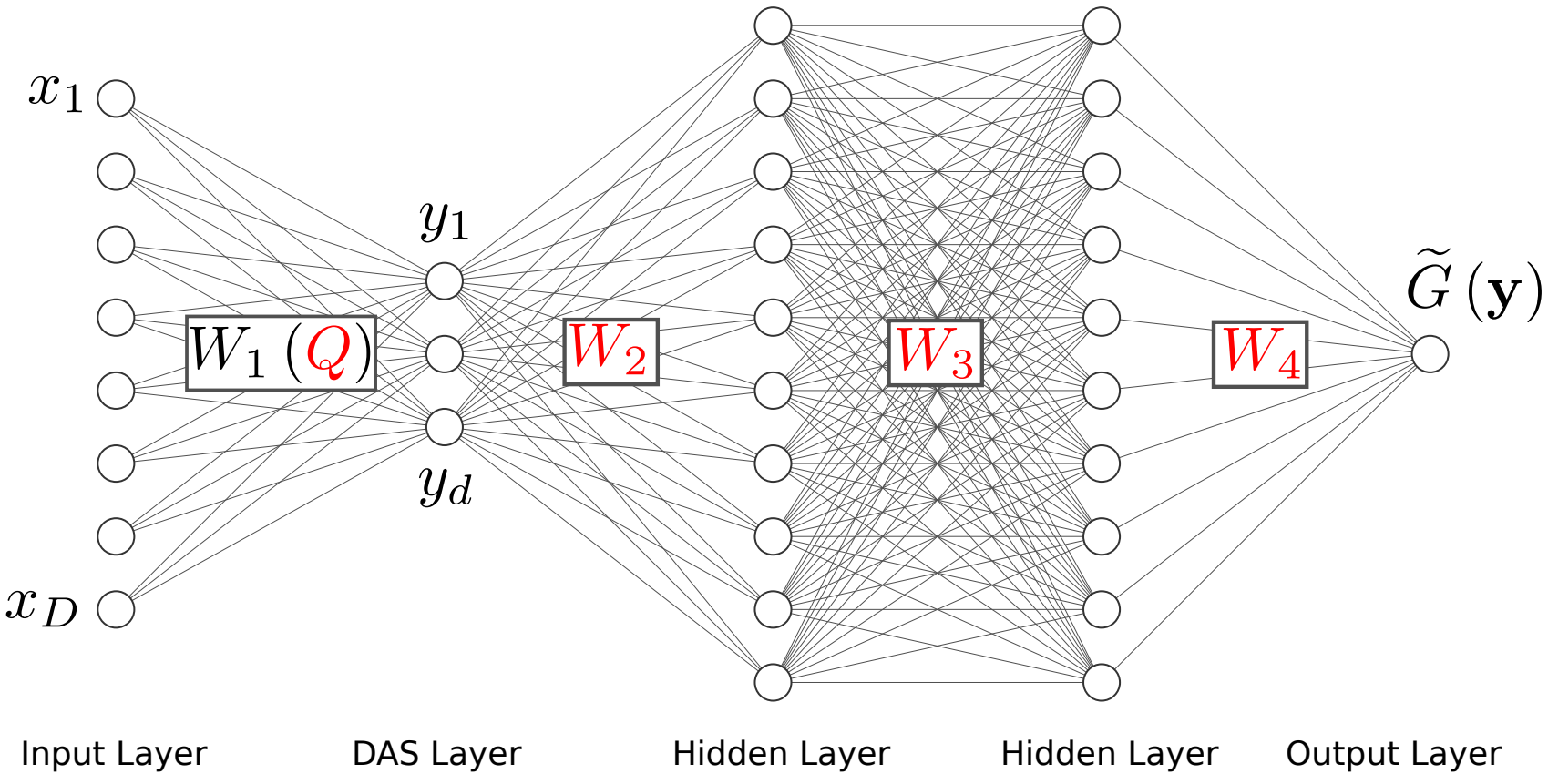
$$\frac{\partial L}{\partial q_{ij}} = \frac{\partial L}{\partial W_1} \frac{\partial W_1}{\partial q_{ij}}$$

- Need the derivatives of Gram-Schmidt vectors: $\frac{\partial \mathrm{w}_i}{\partial \mathrm{q}_j}$
- $\mathrm{w}_i\left(\mathrm{q}_1, \mathrm{q}_2, \cdots, \mathrm{q}_i\right)$ becomes complicated expression for $i > 1$.
- Original DAS article used automatic differentiation
- We used matrix calculus to find exact analytic recurrence relation [4]

[4] https://github.com/wedeling/Gram_Schmidt_Derivatives

Deep active subspaces: surrogate

- Complete picture: $\widetilde{G}(\mathrm{y})$ is neural network from DAS layer onward:



- Unlike adaptive SC, training data is generated from a single, large (Monte Carlo) ensemble.

Results: HIV model

- HIV model, predicts T-cell count over time,

  $\rightarrow$ 7 ODEs 27 uncertain inputs, (uniformly distributed).

  $\rightarrow$ Inputs are normalized to $[-1, 1]$ for DAS network.

$$
\begin{aligned}
\frac{dT}{dt} &= s_1 + \frac{p_1}{C_1 + V} TV - \delta_1 T - (K_1 V + K_2 M_I) T, \\
\frac{dT_I}{dt} &= \psi(K_1 V + K_2 M_I) T + \alpha_1 T_L - \delta_2 T_I - K_3 T_I CTL, \\
\frac{dT_L}{dt} &= (1 - \psi)(K_1 V + K_2 M_I) T - \alpha_1 T_L - \delta_3 T_L, \\
\frac{dM}{dt} &= s_2 + K_4 MV - K_5 MV - \delta_4 M, \\
\frac{dM_I}{dt} &= K_5 MV - \delta_5 M_I - K_6 M_I CTL, \\
\frac{dCTL}{dt} &= s_3 + (K_7 T_I + K_8 M_I) CTL - \delta_6 CTL, \\
\frac{dV}{dt} &= K_9 T_I + K_{10} M_I - K_{11} TV - (K_{12} + K_{13}) MV - \delta_7 V,
\end{aligned}
$$

- Active subspace method (w/ derivatives) has been applied [5]

  $\rightarrow$ use as reference solution.

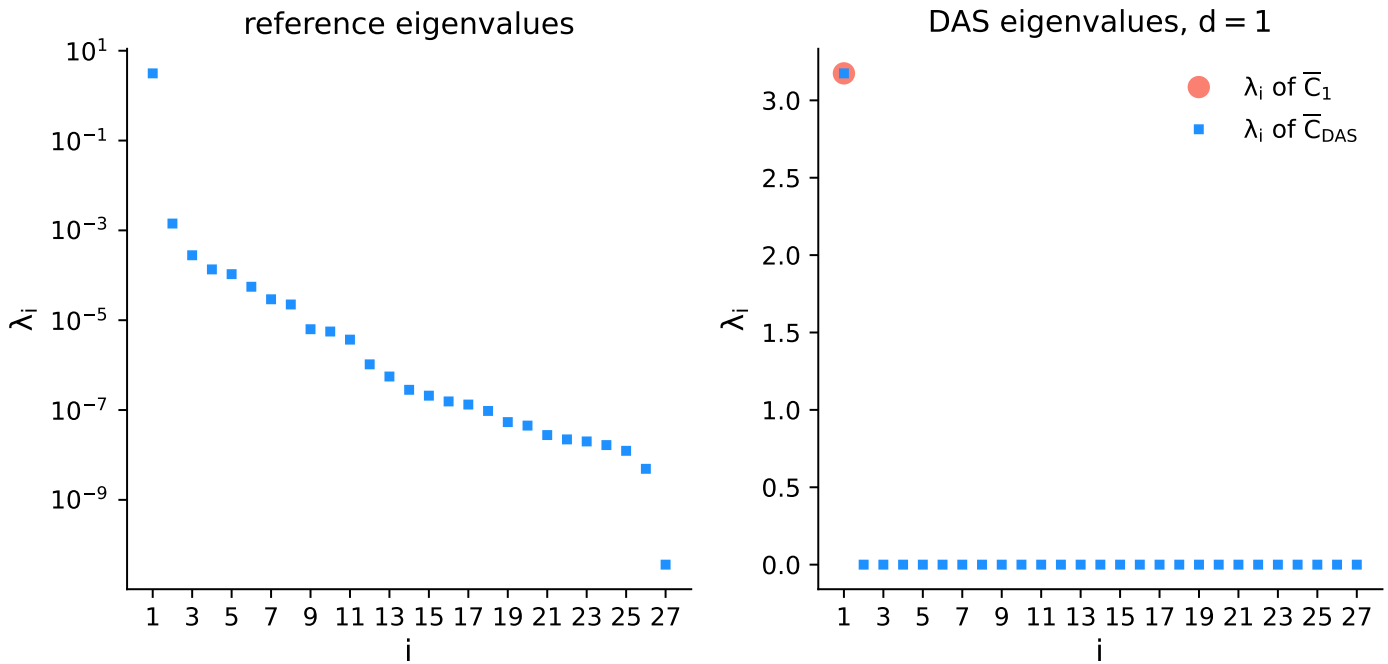
[5] Loudon, T., & Pankavich, S. (2016). Mathematical analysis and dynamic active subspaces for a long term model of HIV. arXiv preprint arXiv:1604.04588.

Results: HIV model

Eigenvalues of:

$$C = \int (\nabla f(x))(\nabla f(x))^T p(x)dx \quad \& \quad C_{DAS} = \int \left(\nabla \widetilde{f}(x)\right)\left(\nabla \widetilde{f}(x)\right)^T p(x)dx$$

f: T-cell count $t = 3400$ days, $\nabla \widetilde{f}$: gradient DAS network (easily available).

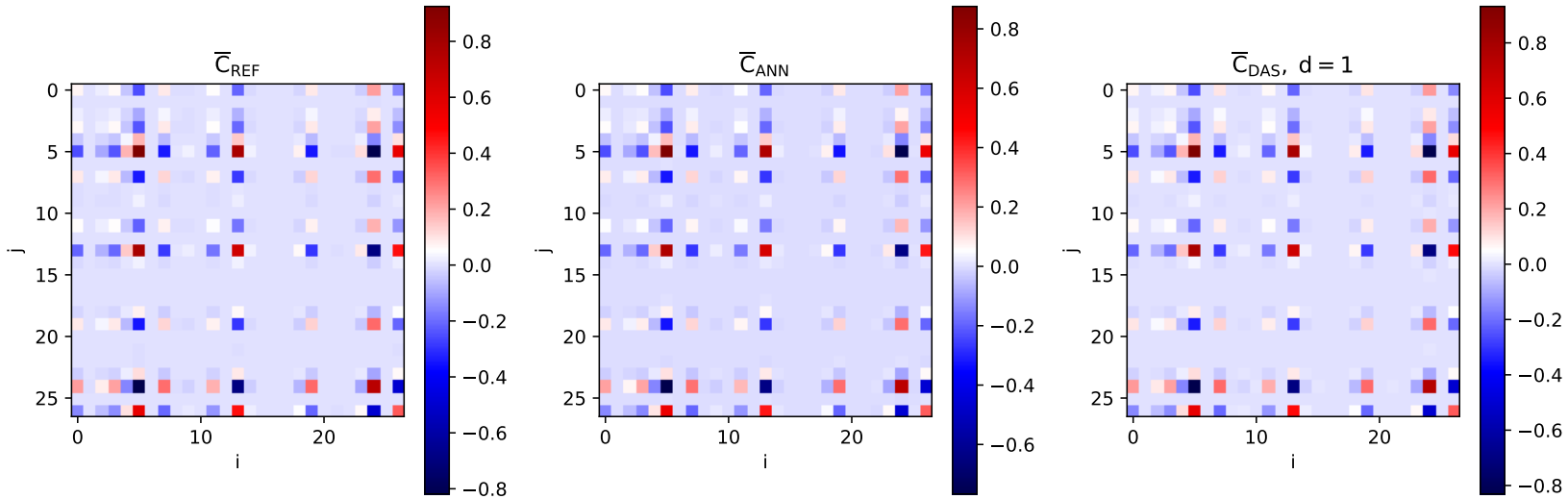


We have no need for C_{DAS} to use DAS method, only done for comparison.

Results: HIV model

Heat map of C from:

- Reference active subspace,
- Standard artificial neural network (ANN, no DAS layer),
- DAS network.

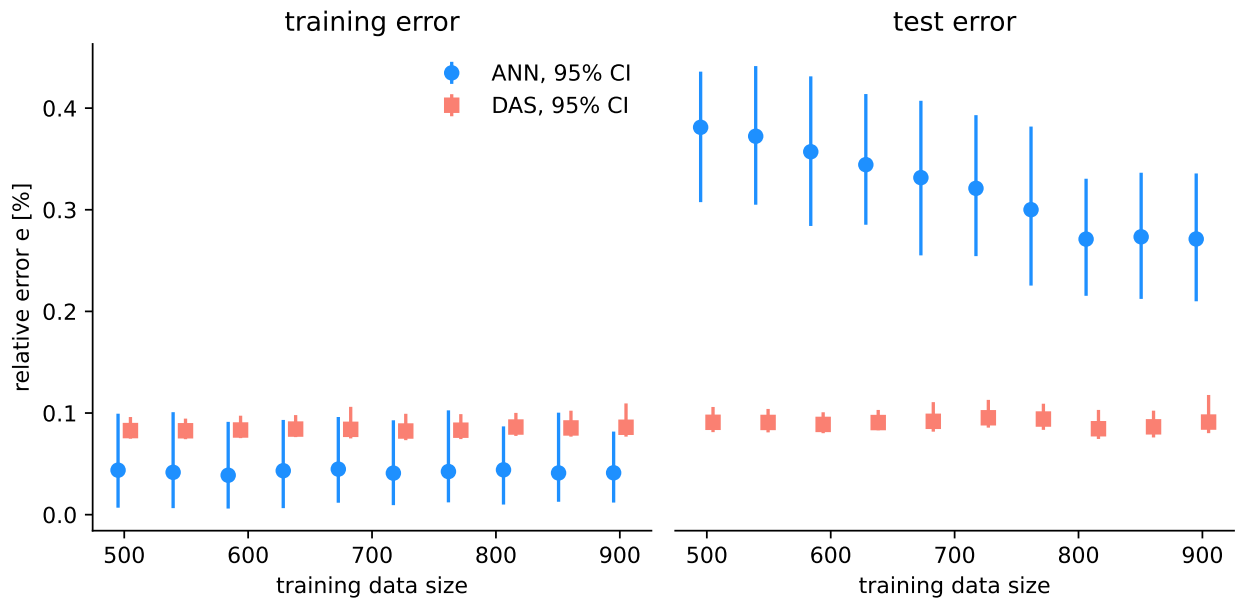


Results: HIV model

ECMA

ANN vs DAS: training and test performance:

- For each training / test split, train 100 replica neural networks,
  $\rightarrow$ compute 95% confidence intervals.
- Using 100 neurons per hidden layer (overfitted):

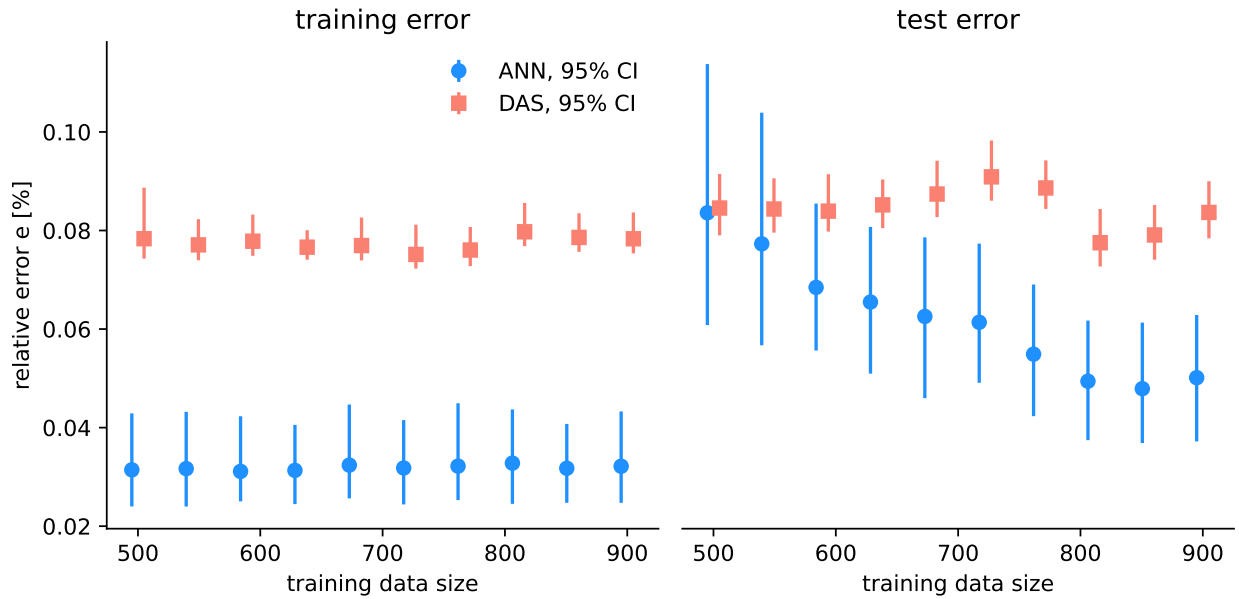


Results: HIV model

ECMA

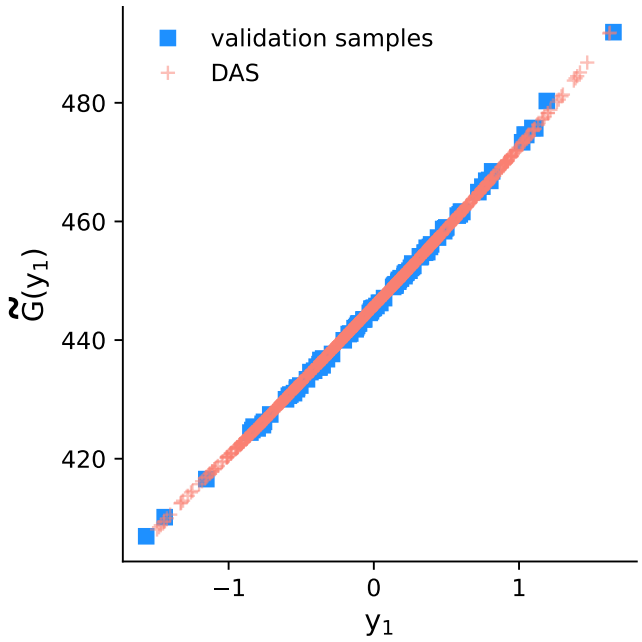
ANN vs DAS: training and test performance:

- For each training / test split, train 100 replica neural networks,
  $\rightarrow$ compute 95% confidence intervals.
- Using 10 neurons per hidden layer :

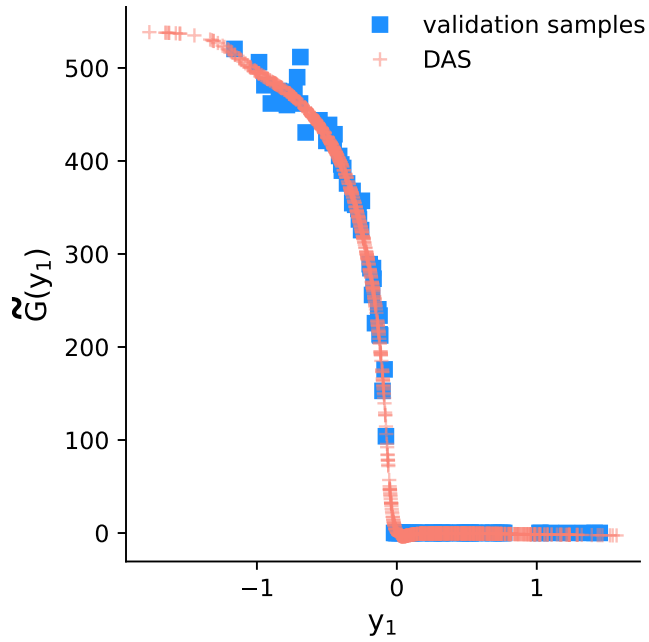


Results: HIV model

Code ($D = 27$) and DAS response ($d = 1$) as function of y_1:



(a) $t = 45$ days



(b) $t = 3400$ days

Results: COVID19 model

- Similar analysis was applied to CovidSim[6].
- 51 inputs parameters, assumed uniform distributions.

[6] Ferguson, N. M., Laydon, D., et al. (2020). Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand.

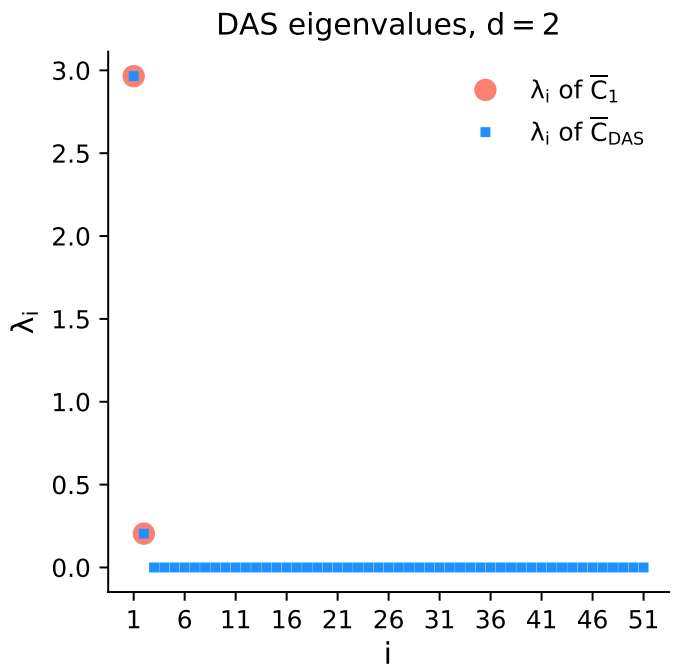
Results: COVID19 model

- Similar analysis was applied to CovidSim[6].
- 51 inputs parameters, assumed uniform distributions.
- EasyVVUQ: MC sampling.
- FabSim3 + QCG PilotJob: data transfer and ensemble execution on PSNC Altair supercomputer.

[6] Ferguson, N. M., Laydon, D., et al. (2020). Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand.

Results: COVID19 model

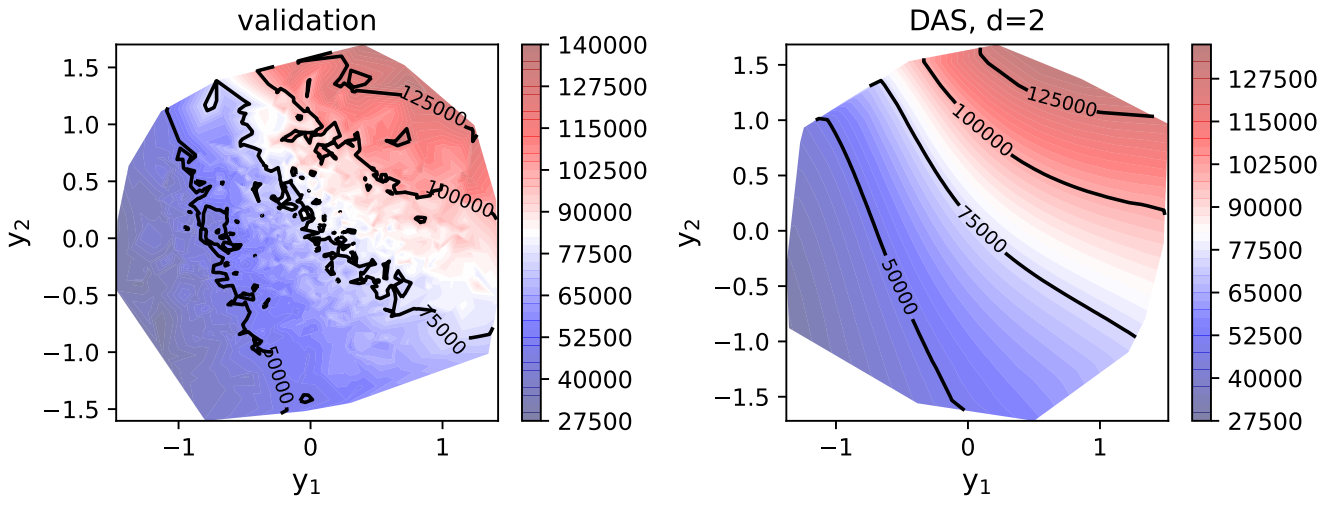
- Similar analysis was applied to CovidSim[6].
- 51 inputs parameters, assumed uniform distributions.
- EasyVVUQ: MC sampling.
- FabSim3 + QCG PilotJob: data transfer and ensemble execution on PSNC Altair supercomputer.
- Chose $d = 2$ as dimension of active subspace:



[6] Ferguson, N. M., Laydon, D., et al. (2020). Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand.

Results: COVID19 model

Code ($D = 51$) and DAS ($d = 2$) contours in (y_1, y_2) space:



$f(\mathrm{x})$: final death count after 2 years.

Conclusion

- (Deep) active subspace method scales better than dimension-adaptive SC (try ≈ 100 inputs next)
- Non-adaptive, requires 1 ensemble, dimension-reduction done as post processing.
- Deep active subspaces require no gradient, approximated active-subspace reference well for HIV model.

Questions?

Preprint: W. Edeling, On the deep active subspace method [7]

To reproduce results:
`https://github.com/wedeling/deep_active_subspace_data`

[7] https://www.researchgate.net/publication/356751004_On_the_deep_active_subspace_method