A Multi-Fidelity Ensemble Kalman Filter with a Machine Learning surrogate model

Jeffrey van der Voort

joint work with: Martin Verlaan Hanne Kekkonen

Delft University of Technology, The Netherlands

October 24th, 2024

The problem

The problem

Ensemble simulations and ensemble data assimilation are used a lot in geosciences...

The problem

Ensemble simulations and ensemble data assimilation are used a lot in geosciences...

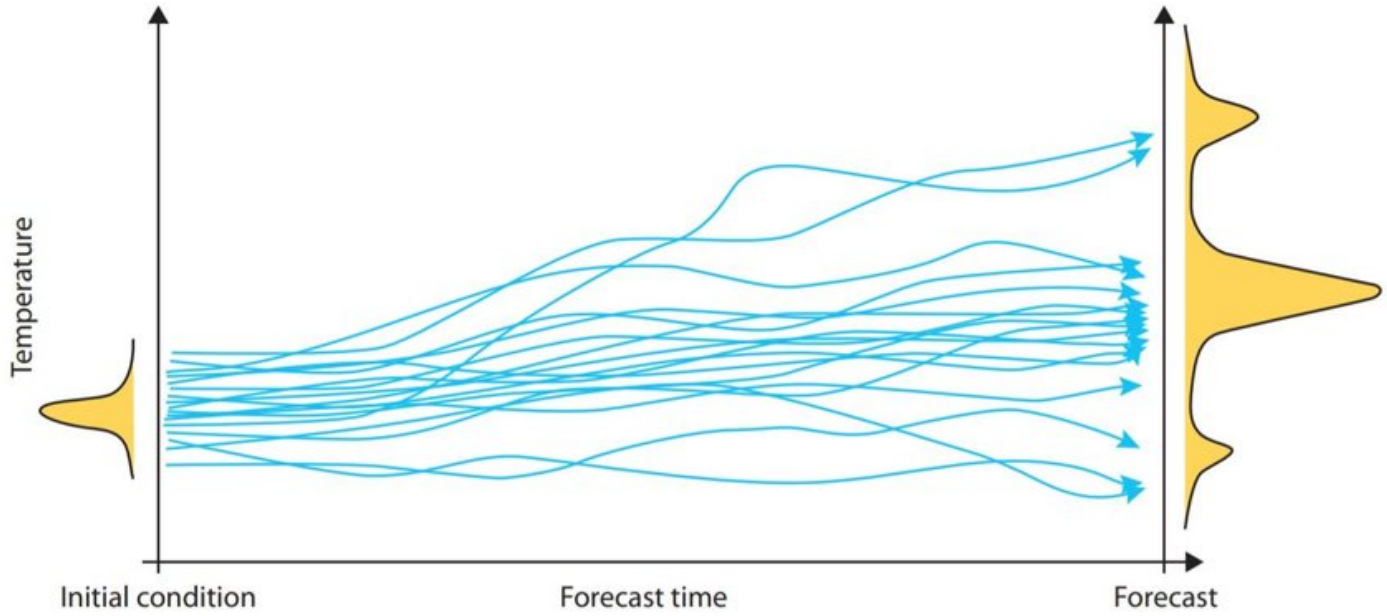

Figure: Figure 1 from Grönquist et al. (2019): *Predicting Weather Uncertainty with Deep Convnets*

The problem

Ensemble simulations and ensemble data assimilation are used a lot in geosciences...

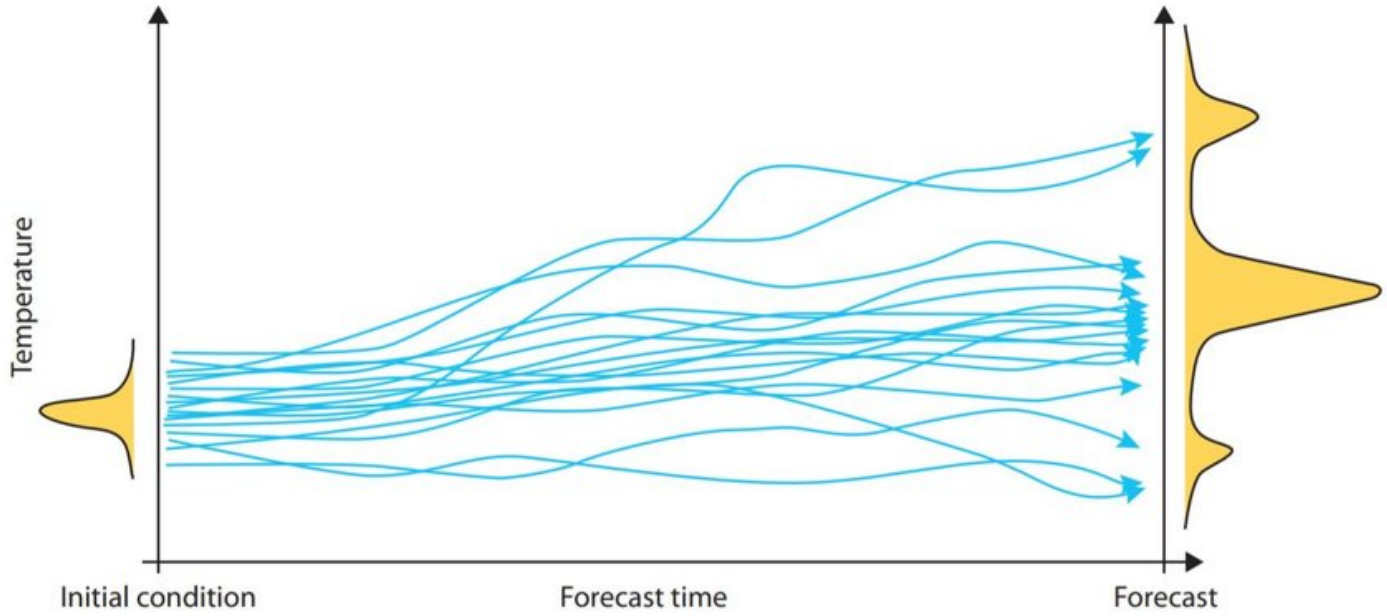

Figure: Figure 1 from Grönquist et al. (2019): *Predicting Weather Uncertainty with Deep Convnets*

...but running ensembles can be very expensive.

The solution!

The solution!

Just completely replace expensive physical model by a cheap ML model!

The solution!

Just completely replace expensive physical model by a cheap ML model!

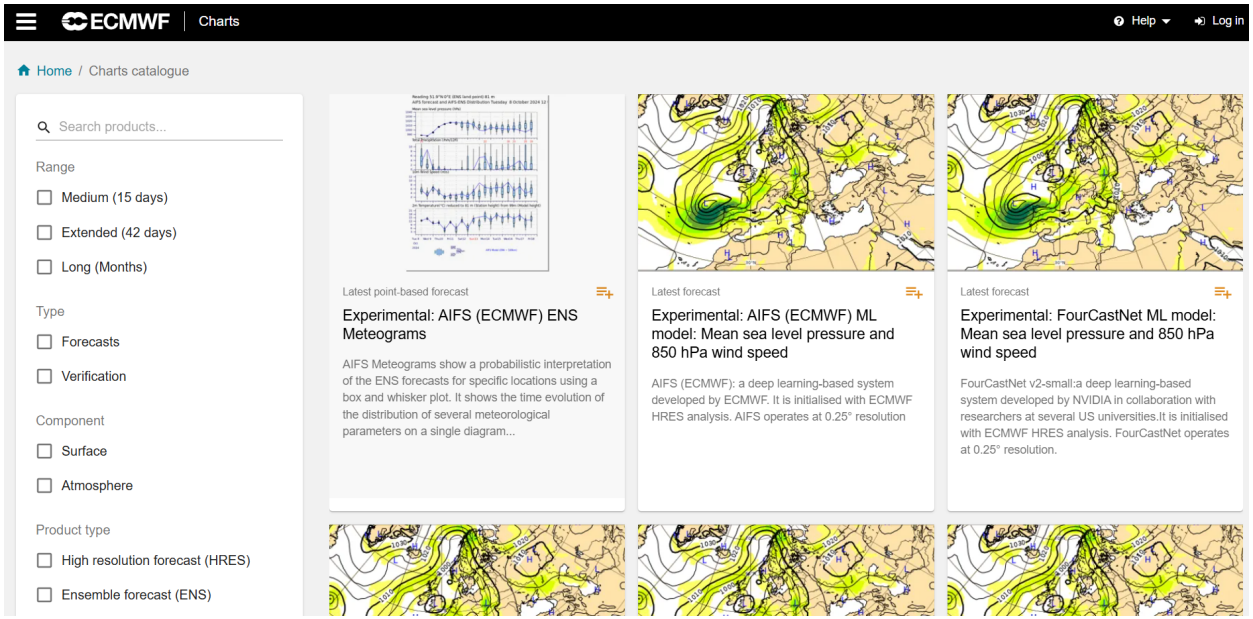

Figure: Experimental ML models, from: `https://charts.ecmwf.int`

The solution!

Just completely replace expensive physical model by a cheap ML model!

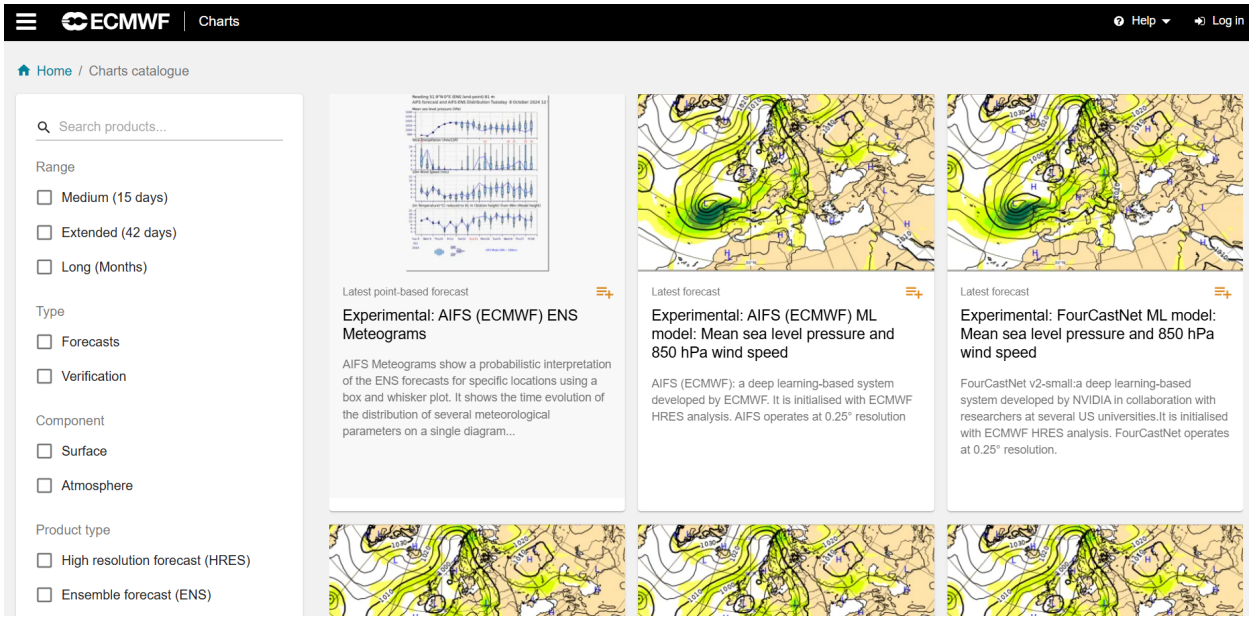

Figure: Experimental ML models, from: `https://charts.ecmwf.int`

...but usually less accurate than the physical model.

Main idea

Main idea

Few expensive, but accurate, full order model runs

Main idea

Few expensive, but accurate, full order model runs

+

Many cheap, but less accurate, surrogate model runs

Main idea

Few expensive, but accurate, full order model runs

+

Many cheap, but less accurate, surrogate model runs

⇒

combined using a Multi Fidelity Ensemble Kalman Filter (MF-EnKF) framework.

Multi-Fidelity approach

Multi-Fidelity approach

Multi-Fidelity estimator is based on the random variable

$$Z = X - \lambda(\hat{U} - U)$$

which is called total variate.

Multi-Fidelity approach

Multi-Fidelity estimator is based on the random variable

$$Z = X - \lambda(\hat{U} - U)$$

which is called total variate.

- Random variable X (state vector), which is expensive to generate samples from. (principal variate)

Multi-Fidelity approach

Multi-Fidelity estimator is based on the random variable

$$Z = X - \lambda(\hat{U} - U)$$

which is called total variate.

- Random variable X (state vector), which is expensive to generate samples from. (principal variate)
- Random variable $\hat{U}$, highly correlated with X and cheap to generate samples from. (control variate)

Multi-Fidelity approach

Multi-Fidelity estimator is based on the random variable

$$Z = X - \lambda(\hat{U} - U)$$

which is called total variate.

- Random variable X (state vector), which is expensive to generate samples from. (principal variate)
- Random variable $\hat{U}$, highly correlated with X and cheap to generate samples from. (control variate)
- Random variable U, same mean as $\hat{U}$, which is used to correct for the bias (ancillary variate)

Multi-Fidelity approach

Multi-Fidelity estimator is based on the random variable

$$Z = X - \lambda(\hat{U} - U)$$

which is called total variate.

- Random variable X (state vector), which is expensive to generate samples from. (principal variate)
- Random variable $\hat{U}$, highly correlated with X and cheap to generate samples from. (control variate)
- Random variable U, same mean as $\hat{U}$, which is used to correct for the bias (ancillary variate)
- λ tuning parameter: determines how large the correction $\hat{U} - U$ is

MF-EnKF

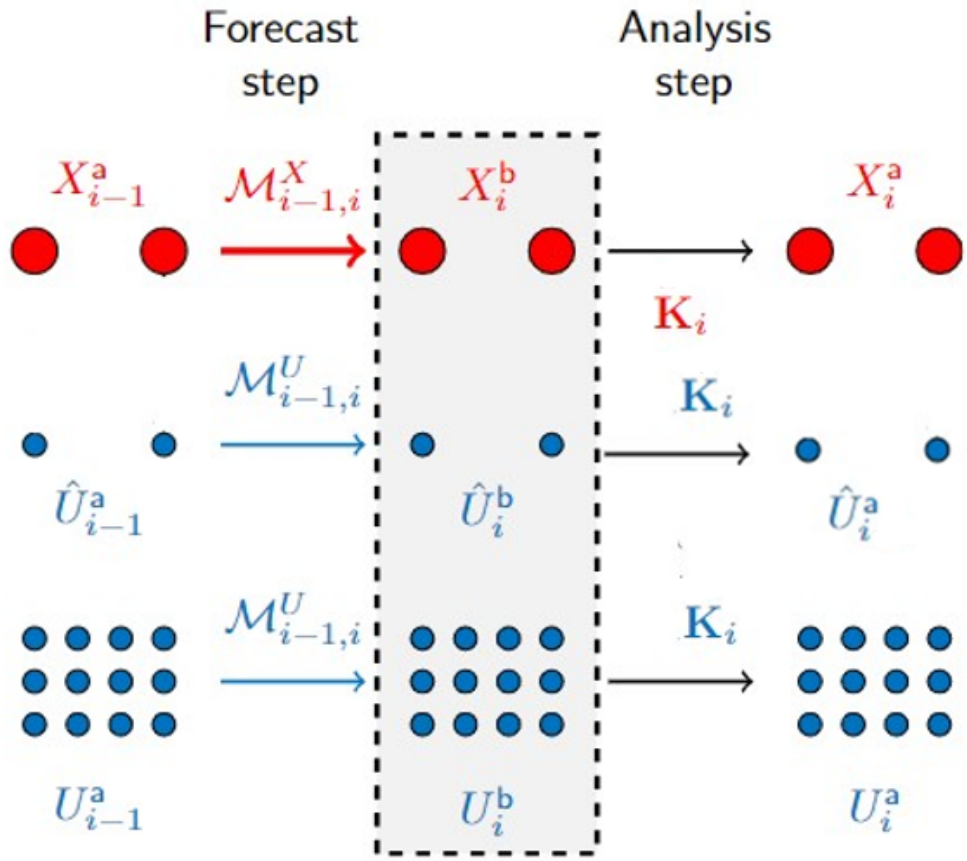

Figure: Figure 4.1 [edited] from Popov et al. (2021): *A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates*

MF-EnKF

Forecast step

$$X_k^{b,(i)} = \mathcal{M}^X(X_{k-1}^{(i)}) \qquad i = 1, \ldots, N_X$$

$$\hat{U}_k^{b,(i)} = \mathcal{M}^U(\hat{U}_{k-1}^{(i)}) \qquad i = 1, \ldots, N_X,$$

$$U_k^{b,(i)} = \mathcal{M}^U(U_{k-1}^{(i)}) \qquad i = 1, \ldots, N_U,$$

MF-EnKF

Forecast step

$$X_k^{b,(i)} = \mathcal{M}^X(X_{k-1}^{(i)}) \qquad i = 1, \ldots, N_X$$

$$\hat{U}_k^{b,(i)} = \mathcal{M}^U(\hat{U}_{k-1}^{(i)}) \qquad i = 1, \ldots, N_X,$$

$$U_k^{b,(i)} = \mathcal{M}^U(U_{k-1}^{(i)}) \qquad i = 1, \ldots, N_U,$$

where

- $\mathcal{M}^X$ the expensive forward model
- $\mathcal{M}^U$ the cheap surrogate model (in this case a Neural Network)
- N_X the number of full runs
- N_U the number of surrogate / ML runs

MF-EnKF

Analysis step

$$X_k^{a,(i)} = X_k^{b,(i)} - K(Y_k - \mathcal{H}(X_k^{b,(i)}) - \varepsilon_k^{X,(i)}), \qquad i = 1, \ldots, N_X$$

$$\hat{U}_k^{a,(i)} = \hat{U}_k^{b,(i)} - K(Y_k - \mathcal{H}(\hat{U}_k^{b,(i)}) - \varepsilon_k^{\hat{U},(i)}), \qquad i = 1, \ldots, N_X$$

$$U_k^{a,(i)} = U_k^{b,(i)} - K(Y_k - \mathcal{H}(U_k^{b,(i)}) - \varepsilon_k^{U,(i)}), \qquad i = 1, \ldots, N_U$$

MF-EnKF

Analysis step

$$X_k^{a,(i)} = X_k^{b,(i)} - K(Y_k - \mathcal{H}(X_k^{b,(i)}) - \varepsilon_k^{X,(i)}), \qquad i = 1, \ldots, N_X$$

$$\hat{U}_k^{a,(i)} = \hat{U}_k^{b,(i)} - K(Y_k - \mathcal{H}(\hat{U}_k^{b,(i)}) - \varepsilon_k^{\hat{U},(i)}), \qquad i = 1, \ldots, N_X$$

$$U_k^{a,(i)} = U_k^{b,(i)} - K(Y_k - \mathcal{H}(U_k^{b,(i)}) - \varepsilon_k^{U,(i)}), \qquad i = 1, \ldots, N_U$$

where

- K the Kalman gain for the total variate, defined as

$$K = \Sigma_{Z_k^b, \mathcal{H}(Z_k^b)} \left(\Sigma_{\mathcal{H}_k(Z_k^b), \mathcal{H}(Z_k^b)} + \Sigma_{\eta_k^Z, \eta_k^Z} \right)^{-1}$$

- Y_k the observation at time t_k
- $\mathcal{H}$ the observation operator
- $\varepsilon_k^{X,(i)}, \varepsilon_k^{\hat{U},(i)}$ and $\varepsilon_k^{U,(i)}$ the observation perturbation terms

MF-EnKF

MF-EnKF

Total analysis

$$Z_k^a = X_k^a - \lambda(\hat{U}_k^a - U_k^a)$$

The total analysis ensemble has mean

$$\mu_{Z_k^a} = \frac{1}{N_X}\sum_{i=1}^{N_X} X_k^{a,(i)} - \lambda\left(\frac{1}{N_X}\sum_{i=1}^{N_X}\hat{U}_k^{a,(i)} - \frac{1}{N_U}\sum_{i=1}^{N_U} U_k^{a,(i)}\right)$$

and covariance

$$\Sigma_{Z_k^a,Z_k^a} = \Sigma_{X_k^a,X_k^a} + \lambda^2\Sigma_{\hat{U}_k^a,\hat{U}_k^a} - \lambda\Sigma_{X_k^a,\hat{U}_k^a} - \lambda\Sigma_{\hat{U}_k^a,X_k^a} + \lambda^2\Sigma_{U_k^a,U_k^a}$$

Test case

Lorenz-96 model

Test case

~~Lorenz-96 model~~

Lorenz-2005 model

Test case: Lorenz-2005

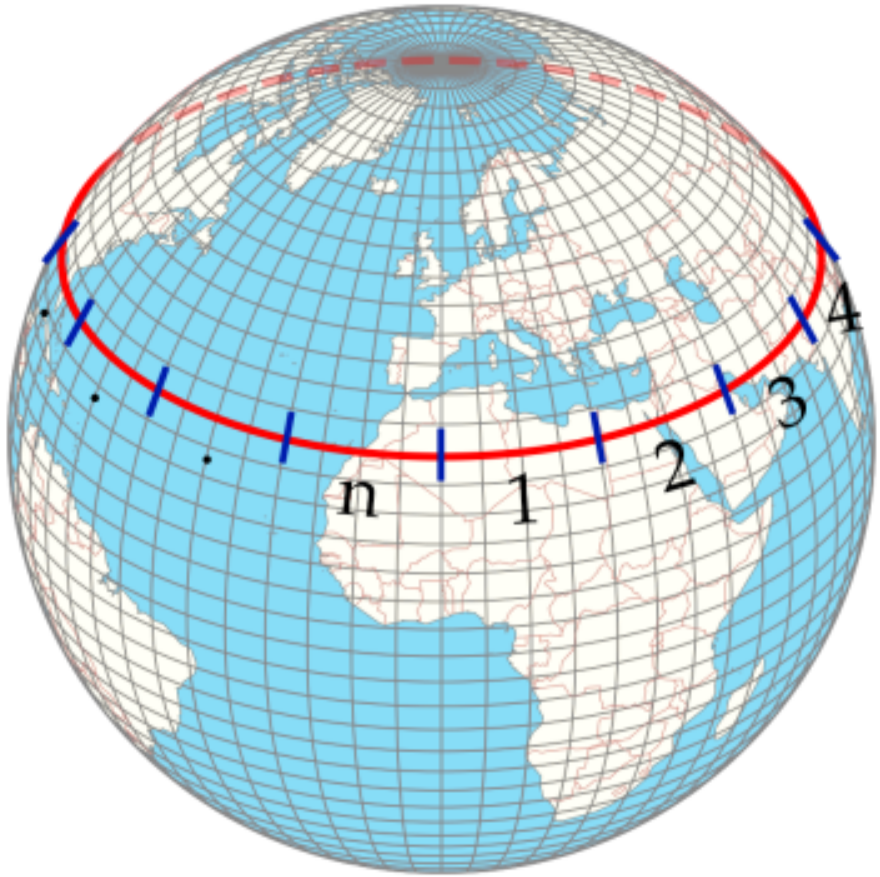

Figure: Atmospheric quantity X (e.g., geopotential height, air temperature) on a circle of constant latitude

Test case: Lorenz-2005

Test case: Lorenz-2005

Type II model from Lorenz (2005): *Designing Chaotic Models* is a generalization of Lorenz-96 that allows for spatial discretization.

$$\frac{dX_i}{dt} = [X, X]_{K,i} - X_i + F, \qquad i = 0, \ldots, n-1$$

Test case: Lorenz-2005

Type II model from Lorenz (2005): *Designing Chaotic Models* is a generalization of Lorenz-96 that allows for spatial discretization.

$$\frac{dX_i}{dt} = [X, X]_{K,i} - X_i + F, \qquad i = 0, \ldots, n-1$$

with K smoothing parameter, F forcing, and

$$[X, X]_{K,i} := \sum_{j=-J}{}' \sum_{k=-J}{}' \left(-X_{i-2K-k} X_{i-K-j} + X_{i-K+j-k} X_{i+K+k}\right)/K^2$$

when K is even, where $\sum'$ is the usual summation with the first and last term divided by 2.

Test case: Lorenz-2005

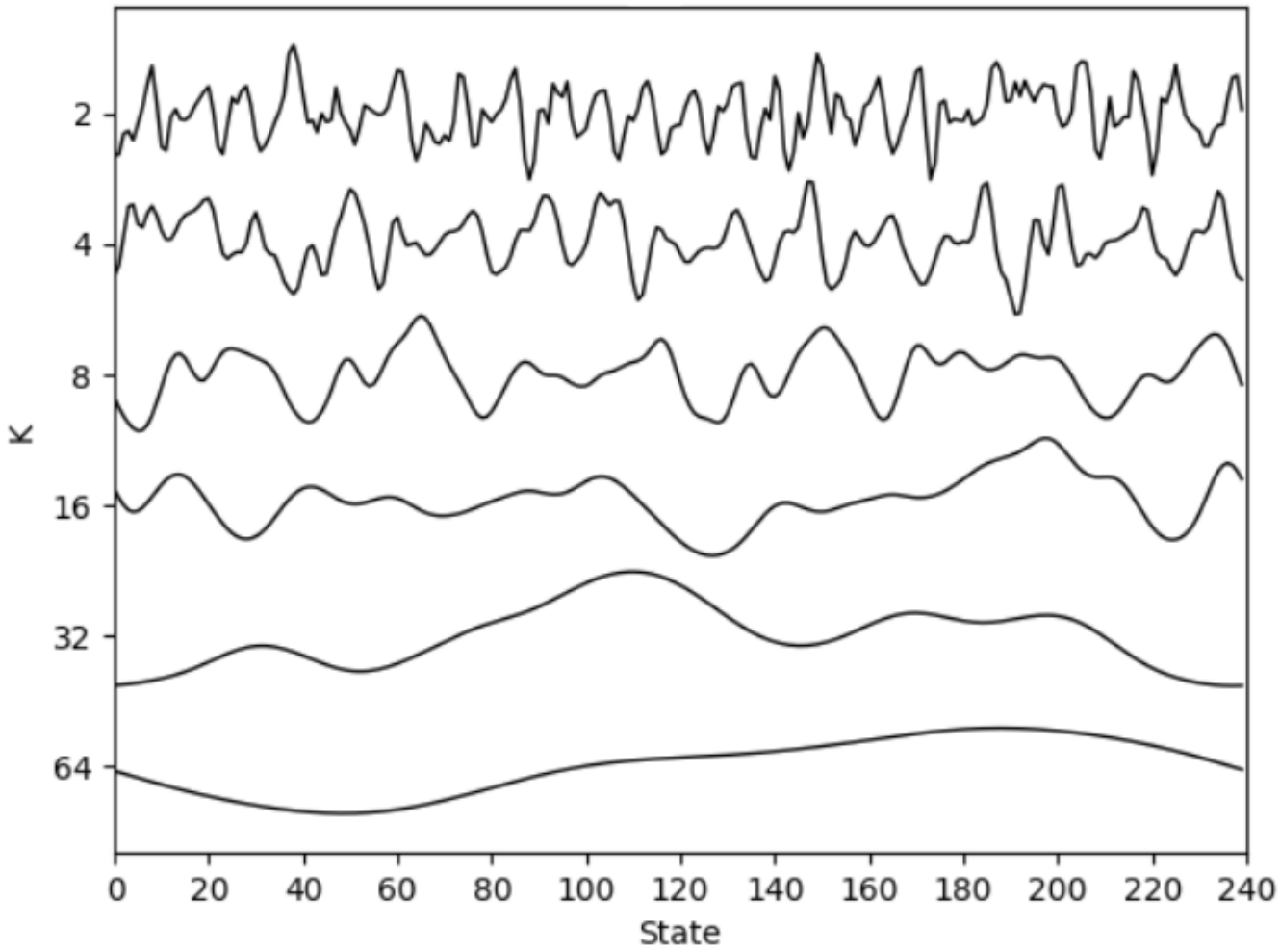

Figure: Influence of smoothing parameter K

Experimental setup

Experimental setup

Truth trajectory. Truth is generated using Lorenz-2005 with $n = 960$, smoothing $K = 32$ and forcing $F = 15$. Spin-up time of 2 years. Model time step: $\Delta t = 0.025$ (corresponds to 3 hours).

Experimental setup

Truth trajectory. Truth is generated using Lorenz-2005 with $n = 960$, smoothing $K = 32$ and forcing $F = 15$. Spin-up time of 2 years. Model time step: $\Delta t = 0.025$ (corresponds to 3 hours).

Observations. Available every $\Delta t = 0.05$ (6 hours) in time and $m = 40$ equidistant locations in space (about 4% of total state). Observation noise: $\sigma_{obs} = 2.0$.

Experimental setup

Truth trajectory. Truth is generated using Lorenz-2005 with $n = 960$, smoothing $K = 32$ and forcing $F = 15$. Spin-up time of 2 years. Model time step: $\Delta t = 0.025$ (corresponds to 3 hours).

Observations. Available every $\Delta t = 0.05$ (6 hours) in time and $m = 40$ equidistant locations in space (about 4% of total state). Observation noise: $\sigma_{obs} = 2.0$.

DA setup. Initial ensemble generated from Gaussian distribution. DA performed for $T = 1000$ time steps, burn-in of 100. Observations are generated with $M = 10$ different seeds, results are averaged. In MF-EnKF, λ is fixed at 0.45

Experimental setup

Truth trajectory. Truth is generated using Lorenz-2005 with $n = 960$, smoothing $K = 32$ and forcing $F = 15$. Spin-up time of 2 years. Model time step: $\Delta t = 0.025$ (corresponds to 3 hours).

Observations. Available every $\Delta t = 0.05$ (6 hours) in time and $m = 40$ equidistant locations in space (about 4% of total state). Observation noise: $\sigma_{obs} = 2.0$.

DA setup. Initial ensemble generated from Gaussian distribution. DA performed for $T = 1000$ time steps, burn-in of 100. Observations are generated with $M = 10$ different seeds, results are averaged. In MF-EnKF, λ is fixed at 0.45

ML part. CNN trained on 2 years of training data.

Comparison with optimally tuned EnKF

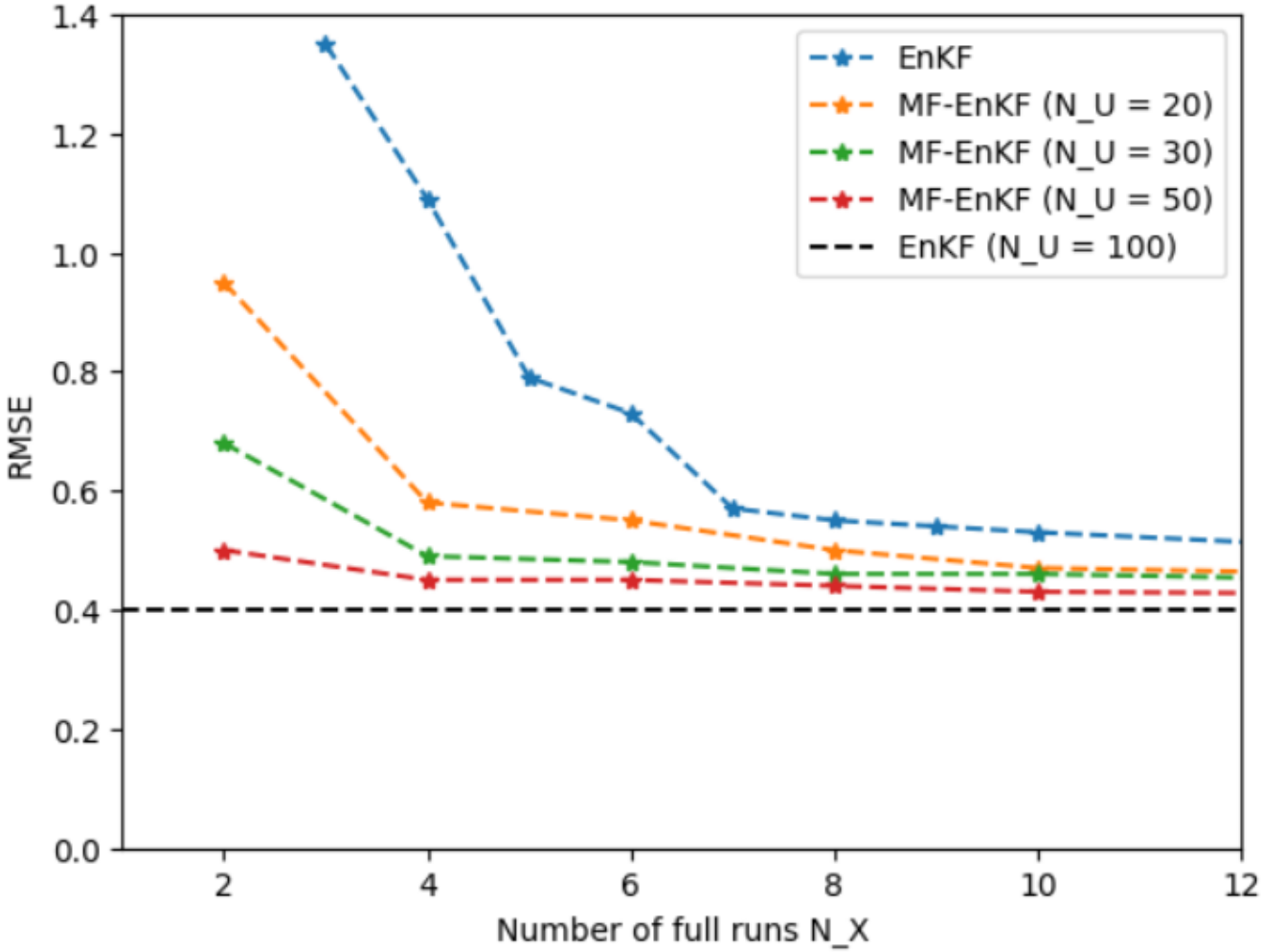

Figure: MF-EnKF versus optimally tuned EnKF

Computational cost

For a fixed computational budget, we can divide between full and surrogate runs. Assume the ML surrogate is 10x faster than the physical model. Assume the budget is 10 physical model runs.

Computational cost

For a fixed computational budget, we can divide between full and surrogate runs. Assume the ML surrogate is 10x faster than the physical model. Assume the budget is 10 physical model runs.

Then we can run MF-EnKF with either

$$N_X = 10, \qquad N_U = 0,$$

Computational cost

For a fixed computational budget, we can divide between full and surrogate runs. Assume the ML surrogate is 10x faster than the physical model. Assume the budget is 10 physical model runs.

Then we can run MF-EnKF with either

$$N_X = 10, \qquad N_U = 0,$$

or

$$N_X = 9, \qquad N_U = 10,$$

Computational cost

For a fixed computational budget, we can divide between full and surrogate runs. Assume the ML surrogate is 10x faster than the physical model. Assume the budget is 10 physical model runs.

Then we can run MF-EnKF with either

$$N_X = 10, \qquad N_U = 0,$$

or

$$N_X = 9, \qquad N_U = 10,$$

or

$$N_X = 8, \qquad N_U = 20,$$

etc.

Computational cost

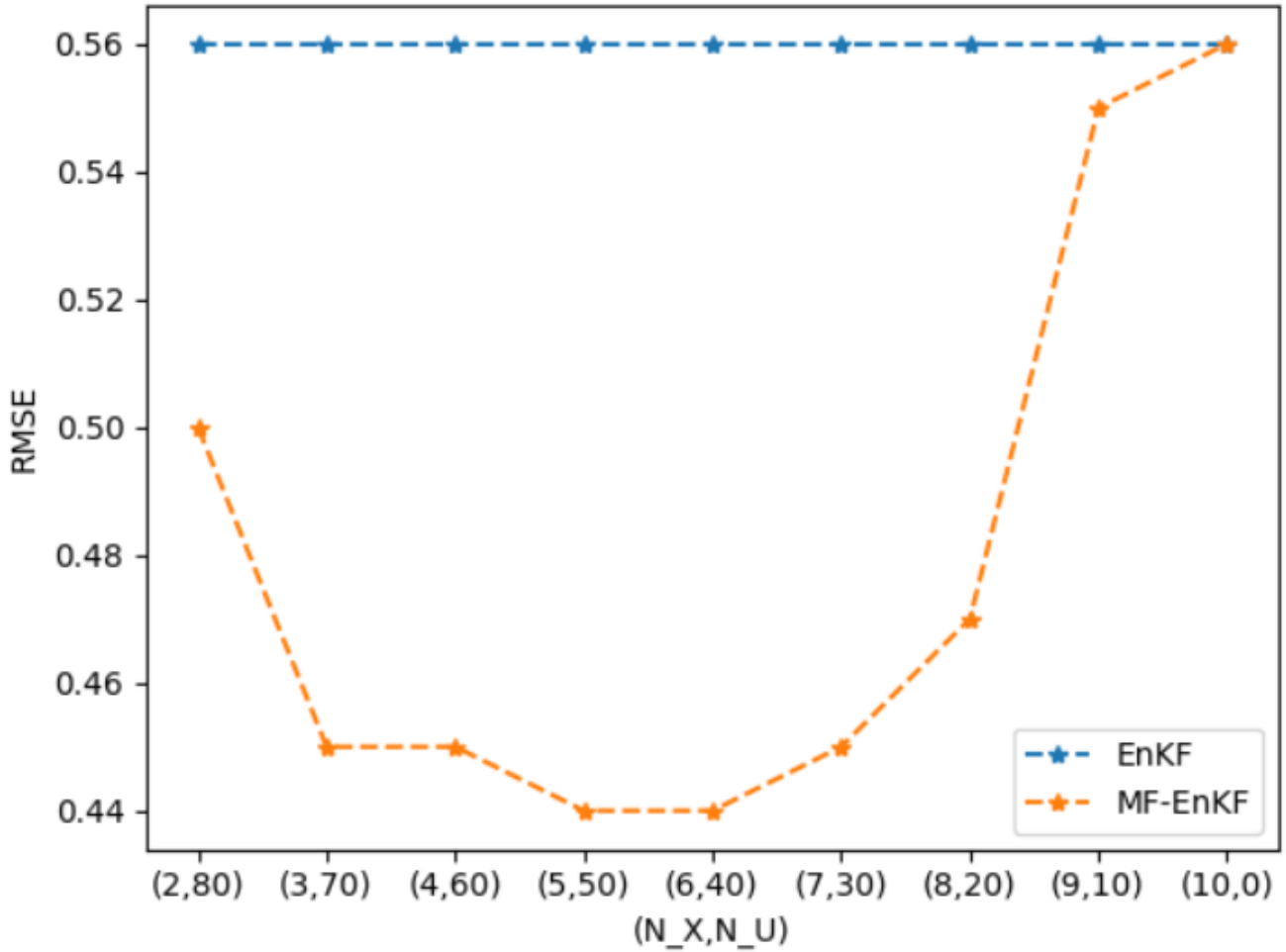

Figure: MF-EnKF vs EnKF for fixed computational cost of $N_X = 10$ full model runs

Comparison with low-res surrogate

We compare NN with different low-resolution surrogate models: m120, m240 and m480. These are generated from Lorenz-2005 with $n = 120$ / $n = 240$ / $n = 480$ and K chosen such that $\frac{n}{K} = 30$ as in the true model.

Comparison with low-res surrogate

We compare NN with different low-resolution surrogate models: m120, m240 and m480. These are generated from Lorenz-2005 with $n = 120$ / $n = 240$ / $n = 480$ and K chosen such that $\frac{n}{K} = 30$ as in the true model.

Model	Forecast RMSE
NN	0.11
m120	0.32
m240	0.09
m480	0.02

Table: RMSE for different surrogate choices

Comparison with low-res surrogate

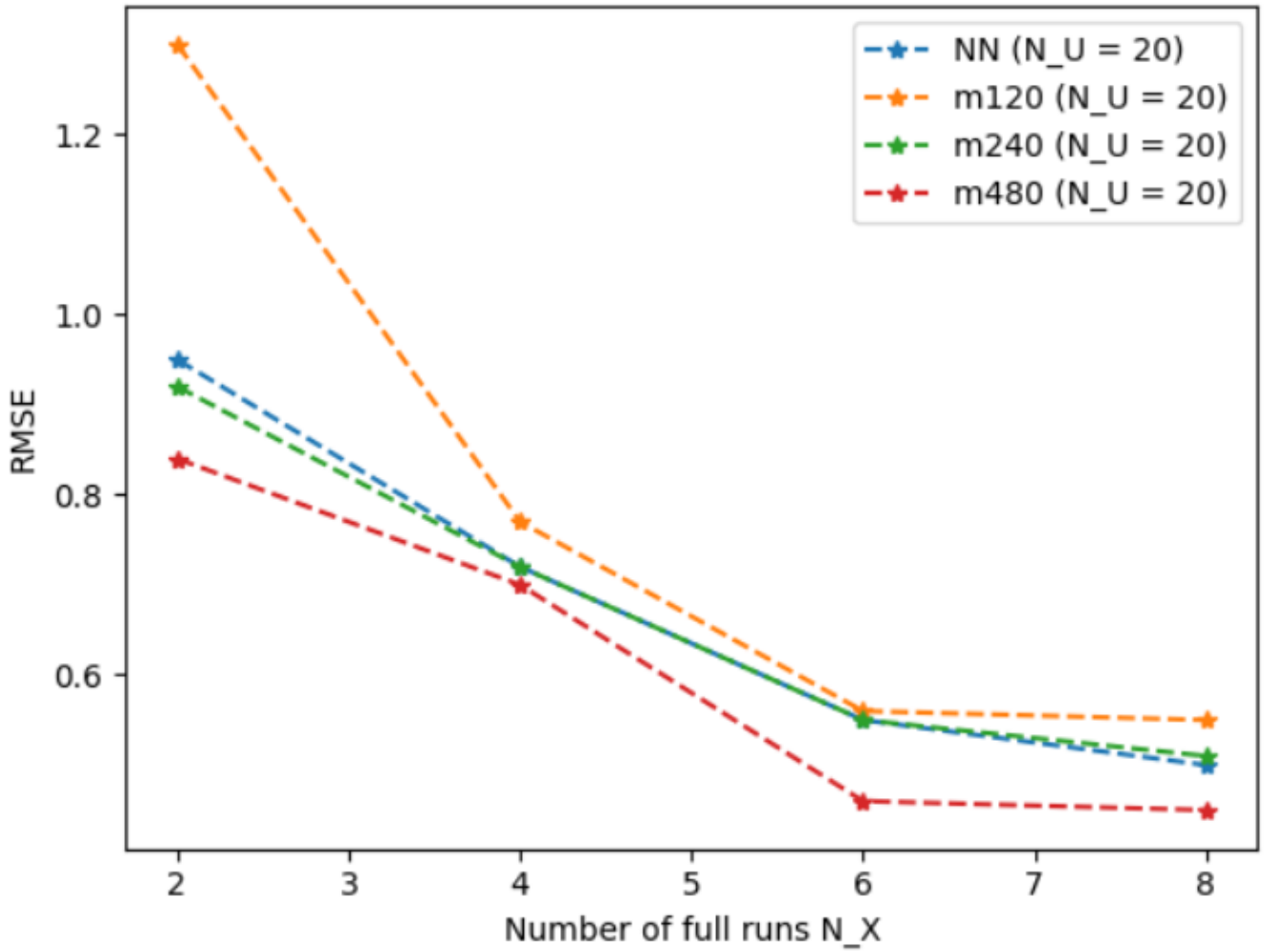

Figure: NN versus low-res models

Conclusion and Discussion

Conclusions

- MF-EnKF outperforms EnKF for fixed number of full runs
- MF-EnKF outperforms EnKF for fixed computational budget
- NN surrogate has similar performance as low-res surrogate, seems linked to quality of the surrogate

Conclusion and Discussion

Conclusions

- MF-EnKF outperforms EnKF for fixed number of full runs
- MF-EnKF outperforms EnKF for fixed computational budget
- NN surrogate has similar performance as low-res surrogate, seems linked to quality of the surrogate

Future

- Redo low-res surrogate experiments for the QG model setting
- Improve NN architecture to surpass low-res model
- Find a way to estimate λ or optimal N_X/N_U combination
- Apply to more realistic problem

Thank you for your attention

References

- Bi, K. et al. (2023). Accurate medium-range global weather forecasting with 3D neural networks. *Nature (619)*
- Bonavita, M. (2024). On some limitations of current machine learning weather prediction models. *Geophysical Research Letters (51)*
- Peherstorfer, B. et al. (2018). Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. *SIAM Review (60)*
- Popov, A. et al. (2021). A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates. *SIAM Journal on Scientific Computing (43)*

Influence of λ

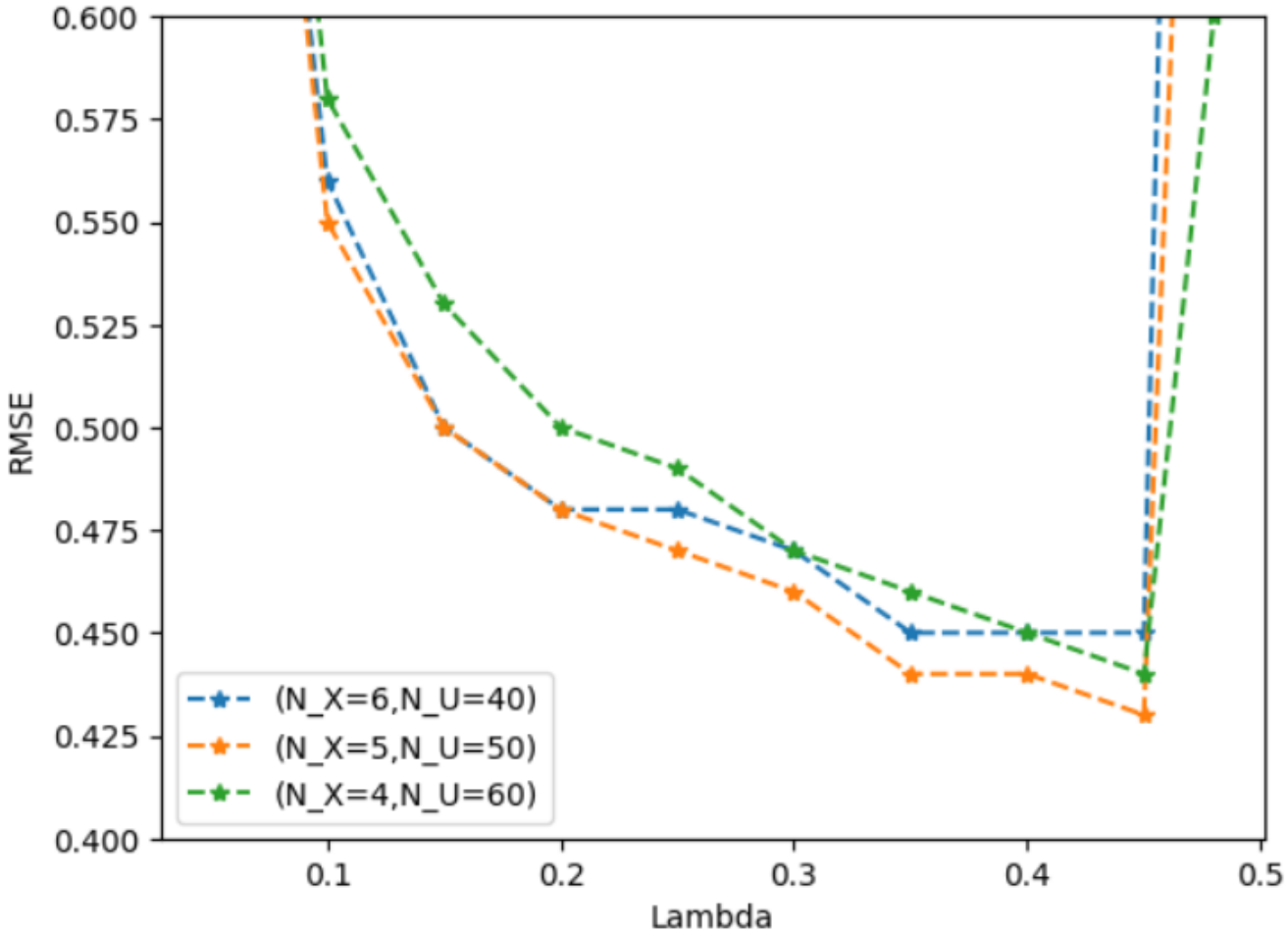

Figure: Influence of λ for different N_X and N_U choices