Machine Learning for Scientific Applications - The Good, the Bad and the Improved

CosmoClub, Zürich
May 11, 2020

Vanessa Böhm

BERKELEY CENTER *for* COSMOLOGICAL PHYSICS

https://github.com/VMBoehm

Outline

PART 1) Machine Learning for Scientific Applications

(Deep) ML methods are becoming increasingly popular, but are often not designed with scientific applications in mind.

- Generative Models: Power, Pitfalls and Improvements
    - Realistic Sample Generation
    - Outlier Detection
    - Probabilistic Reconstruction
- How deadly is COVID-19 - a generative model based approach

PART 2) ML inspired approaches

The indirect uses of deep learning - automatic differentiation.

- MADLens - a madly fast and differentiable lensing simulator.

(Deep) Generative Models

… (neural network based) models that learn the probability distribution of data and are able to sample from it.

I’m going to introduce

- Variational Auto-Encoder
- Normalizing Flows

I’m *not* going to talk about

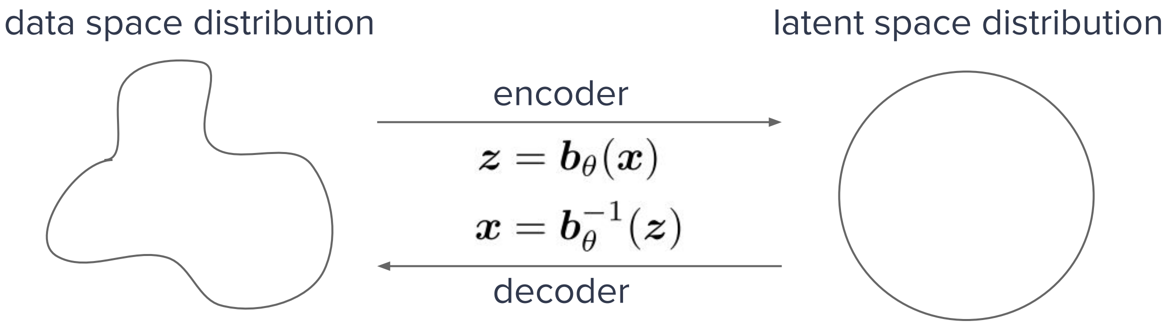
- Generative Adversarial Networks (GANs)

Deep Generative Models

... learn the probability distribution of data and are able to sample from it.

Normalizing Flows

e.g. RealNVP (Dinh et al. 2019), Glow (Kingma et al 2018), MAF (Papamakarios 2017), NSF (Durcan 2019), ...



$$\boldsymbol{z} = \boldsymbol{b}_\theta(\boldsymbol{x})$$

$$\boldsymbol{x} = \boldsymbol{b}_\theta^{-1}(\boldsymbol{z})$$

density estimation : $\ln p_\theta(\boldsymbol{x}) = \ln q(\boldsymbol{z}) + \ln |\nabla_{\boldsymbol{x}} \boldsymbol{b}_\theta(\boldsymbol{x})|$

powers:

- direct density estimation
- bijectivity

weaknesses:

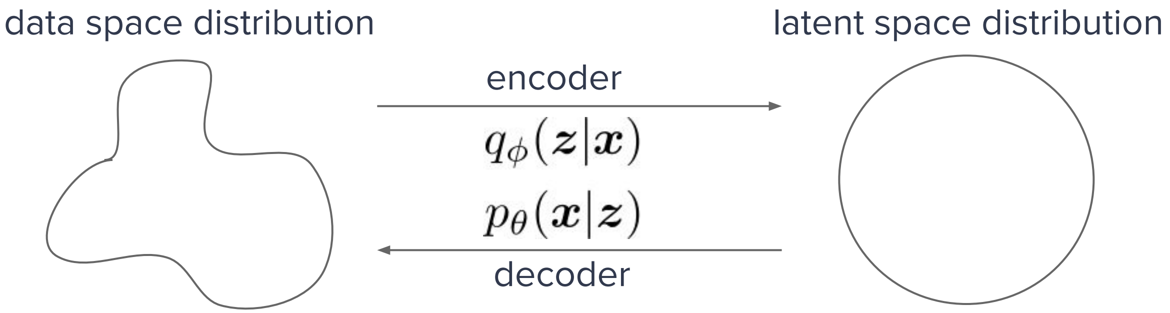
- restrictions on architecture
- scaling with data dimensionality

Deep Generative Models

… learn the probability distribution of data and are able to sample from it.

Variational Auto-Encoder (VAE)

Kingma 2013, Rezende 2014, + countless variants


data space distribution
latent space distribution
encoder
$q_\phi(\boldsymbol{z}|\boldsymbol{x})$
$p_\theta(\boldsymbol{x}|\boldsymbol{z})$
decoder

density estimation :

$$\begin{aligned}\ln p_\theta(\boldsymbol{x}) &= \ln \int d\boldsymbol{z}\, p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) \\ &\geq \mathbb{E}_{q_\phi(z|x)}\left[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right]\end{aligned}$$

powers:

- powerful architectures
- dimensionality reduction
- scalability

weaknesses:

- optimization on lower bound
- posterior approximation
- posterior collapse
- sample quality

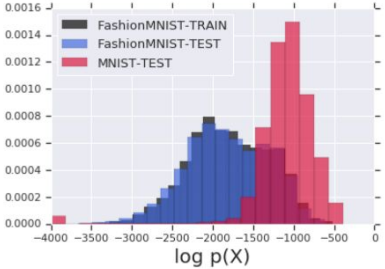
Generative Models for Scientific Applications

Powers

- density estimators that capture otherwise intractable distributions
- sampling from complex distributions
- efficiency/computational tractability
- unsupervised learning and conditional models

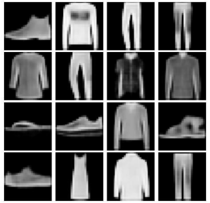
Applications

- (fast) artificial, complex data generation
- probabilistic outlier detection
- probabilistic inference/ data reconstruction

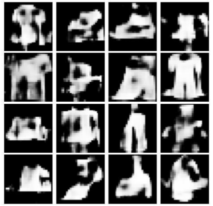


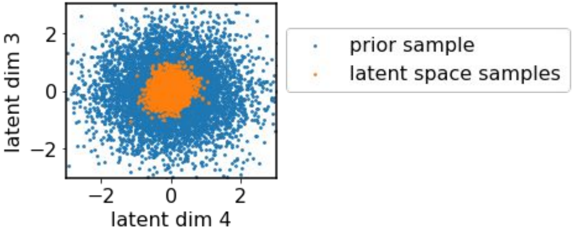
Nalisnick et al. 2019

reconstructions



samples



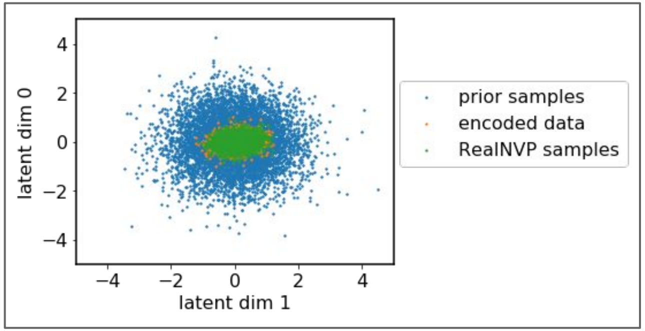


Problems

- have been shown to fail miserably in outlier detection
- often require lots of fine tuning to perform specific tasks well (e.g. β-annealing for VAE sample quality)
- Don’t scale well to high dims (normalizing flows)

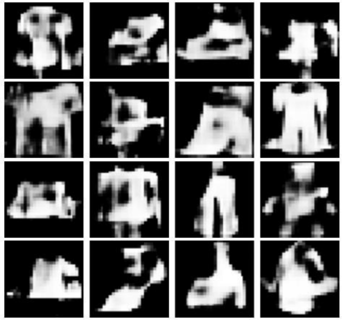
Sample Quality

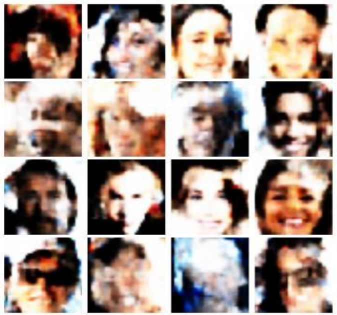
A simple fix



1. Train VAE on data (note that we can even drop the V!-> AE)
2. Fit normalizing flow to the encoded data distribution
3. The new encoder is VAE encoder+ flow encoder, the new decoder is flow Decoder + VAE Decoder

Flow maps samples to the encoded space of the VAE, the VAE decoder can make sense out of these samples



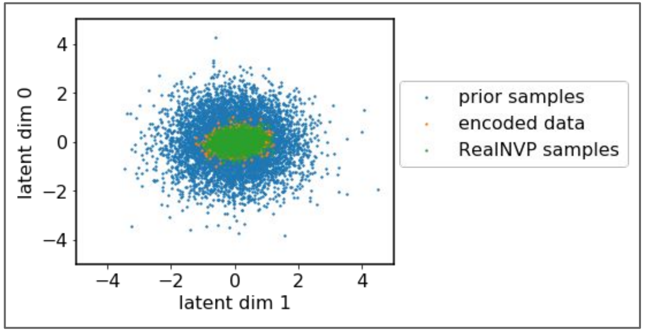


VB, F. Lanusse, U. Seljak 2019, BDL@NeurIPS, 1910.10046

https://github.com/bccp/DeepUQ https://github.com/VMBoehm/vae

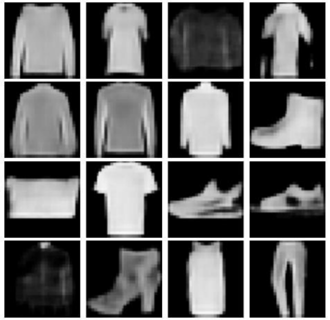
Sample Quality

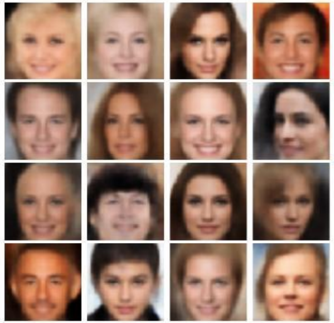
A simple fix



1. Train VAE on data (note that we can even drop the V!-> AE)
2. Fit normalizing flow to the encoded data distribution
3. The new encoder is VAE encoder+ flow encoder, the new decoder is flow Decoder + VAE Decoder

Flow maps samples to the encoded space of the VAE, the VAE decoder can make sense out of these samples





- easy to train (AE+ low dim. flow) ✓
- very little tuning ✓
- high sample quality ✓

VB, F. Lanusse, U. Seljak 2019, BDL@NeurIPS, 1910.10046

https://github.com/bccp/DeepUQ https://github.com/VMBoehm/vae

Probabilistic Auto-Encoder

VB, U. Seljak 2019, submitted to ICML

IDEA: Give up on Variational Inference, estimate density with MAP+Laplace

Step 1: Train a model with encoder/decoder to maximize Maximum of the Posterior (MAP)

$$\boxed{\arg\max_{\theta,\phi}\{\ln p_{n\theta}(\boldsymbol{x}|\boldsymbol{g}_\theta(\boldsymbol{f}_\phi(\boldsymbol{x})),\sigma)+\ln p(\boldsymbol{f}_\phi(\boldsymbol{x}))\}}$$

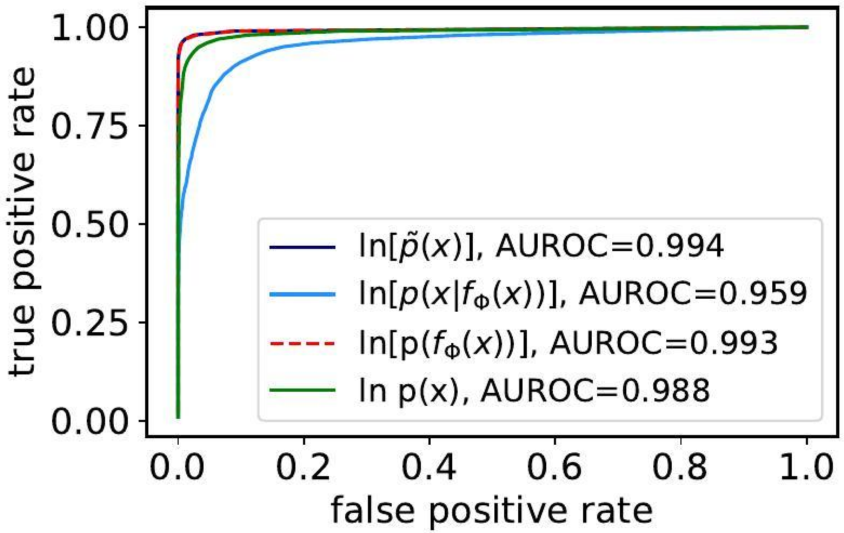
Step 3: Solve marginal integral for data likelihood with Laplace approximation

$$\boxed{p_\theta(\boldsymbol{x})=\int d\boldsymbol{z}\,p(\boldsymbol{z})p_{n\theta}(\boldsymbol{x}|\boldsymbol{z},\sigma)}$$

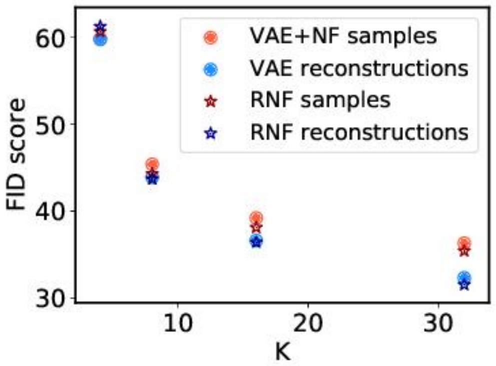
$$\boxed{\ln p(\boldsymbol{x})\approx\ln p(\boldsymbol{x}|\boldsymbol{g}_\theta(\boldsymbol{z}),\boldsymbol{\sigma})+\ln p(\boldsymbol{z})+\frac{1}{2}\ln\det\boldsymbol{\Sigma}_z}$$

VB, U. Seljak 2019, submitted to ICML

PAE Benchmarks



- state of the art sample quality ✓
- reliable outlier detection ✓
- probabilistic ✓
- easy to train ✓
- numerically stable ✓
- intuitive interpretation ✓
- natural regularization ✓



Probabilistic Data Reconstruction

Task:
We want to reconstruct high dimensional corrupted data samples and get reliable uncertainty estimates on the reconstruction.

Probabilistic Inference

Possibility 1: Classical Bayesian Inference

✓ uses all available (physical) knowledge

✓ control over what is put in

✓ versatile and exact

❌ intractable uncertainty quantification and optimization (curse of dimensionality)

❌ lack of priors that optimally capture uncorrupted data distribution

observed data

noise

$$\boxed{y = Ax + n}$$

corruption operator

uncorrupted data/signal

$$p(x|y) = p(y|x)p(x)$$

high dimensional posterior

unknown prior/ data distr.

Probabilistic Inference

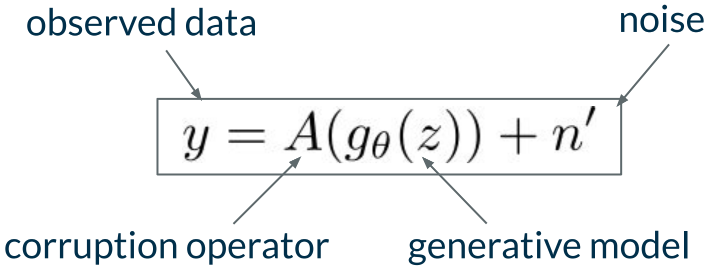
<u>Possibility 2: algorithms based on generative models (mostly VAEs)</u>

Rezende 2014, Ivanov et al 2019, ...

✓ learn correct uncorrupted data distribution (better prior)
✓ potentially computationally more tractable (dimensionality reduction/amortization)
❌ approximate posterior (often amortized)
❌ usually need to be retrained for different corruption types
❌ loss of interpretability

Deep Uncertainty Quantification (DeepUQ)

VB, F. Lanusse, U. Seljak 2019, BDL@NeurIPS, 1910.10046



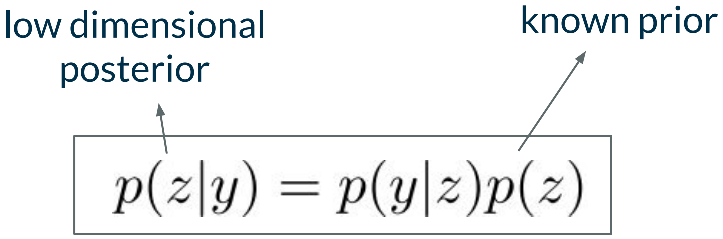
observed data

noise

$$y = A(g_\theta(z)) + n'$$

corruption operator

generative model



low dimensional posterior

known prior

$$p(z|y) = p(y|z)p(z)$$

1. Train a VAE+Flow model on uncorrupted data
2. discard the encoder
3. Replace x by it's generative process
4. The new, exact prior distribution is Gaussian

- optimization/ uncertainty quantification in low dimensions ✓
- exact prior ✓
- versatile, only need to train model once on uncorrupted data ✓
- uses all available physical knowledge ✓

VB, F. Lanusse, U. Seljak 2019, BDL@NeurIPS, 1910.10046

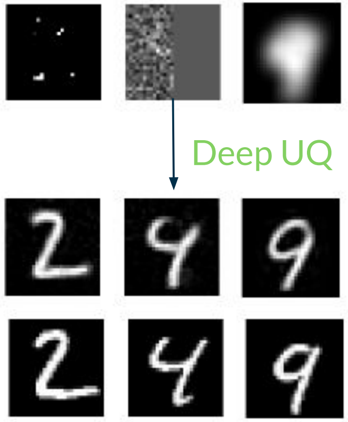
Deep Uncertainty Quantification

Reconstruction

corrupted input data

VB, F. Lanusse, U. Seljak 2019, BDL@NeurIPS, 1910.10046

Deep Uncertainty Quantification

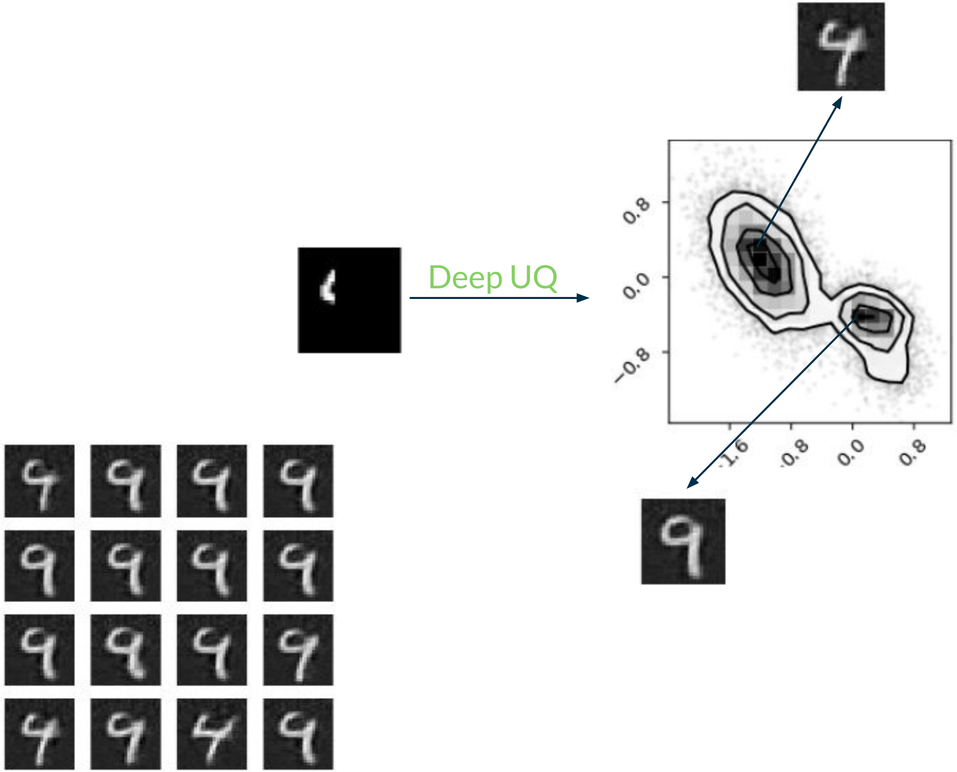


corrupted input data

Deep UQ

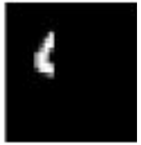
reconstructions

underlying truth



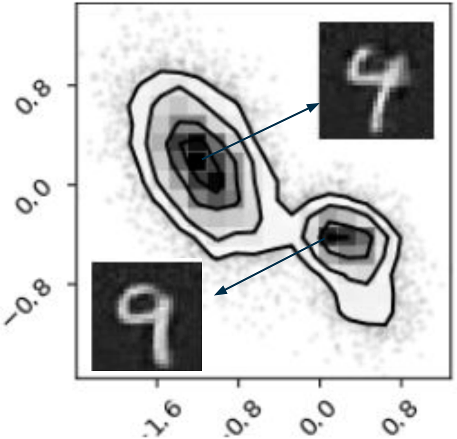
Deep UQ

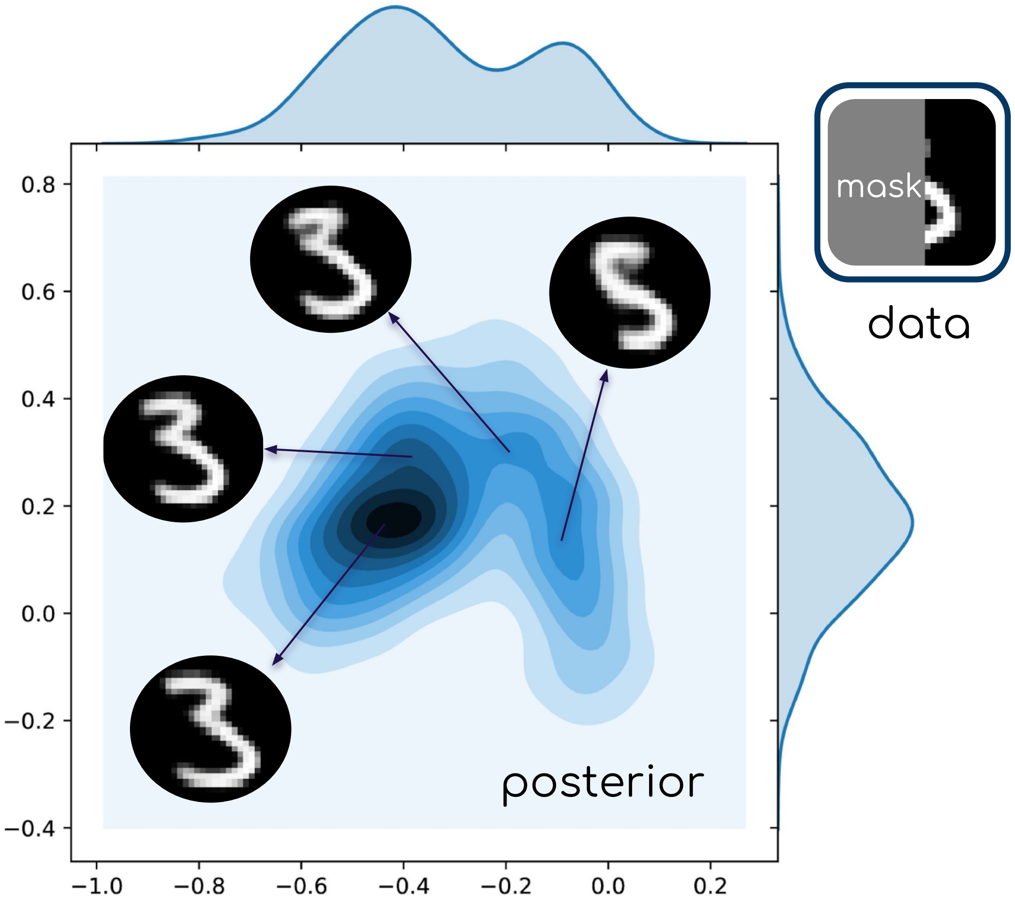
https://github.com/bccp/DeepUQ



data

posterior




0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
posterior
mask
data

Application of a Generative Model: How deadly is COVID-19?

together with **Chirag Modi,** Simone Ferraro, Uros Seljak, George Stein (https://doi.org/10.1101/2020.04.15.20067074)

How deadly is COVID-19?

requires knowledge of two numbers:

1) total number of deaths
2) total number of infections

a lot of discussion has focused on getting 2) right (e.g. antibody tests)

Are we sure that all deaths have been counted?
Can we find a dataset that determines 1) independently of testing?

Application of a Generative Model: How deadly is COVID-19?

together with **Chirag Modi,** Simone Ferraro, Uros Seljak, George Stein (https://doi.org/10.1101/2020.04.15.20067074)

How deadly is COVID-19?

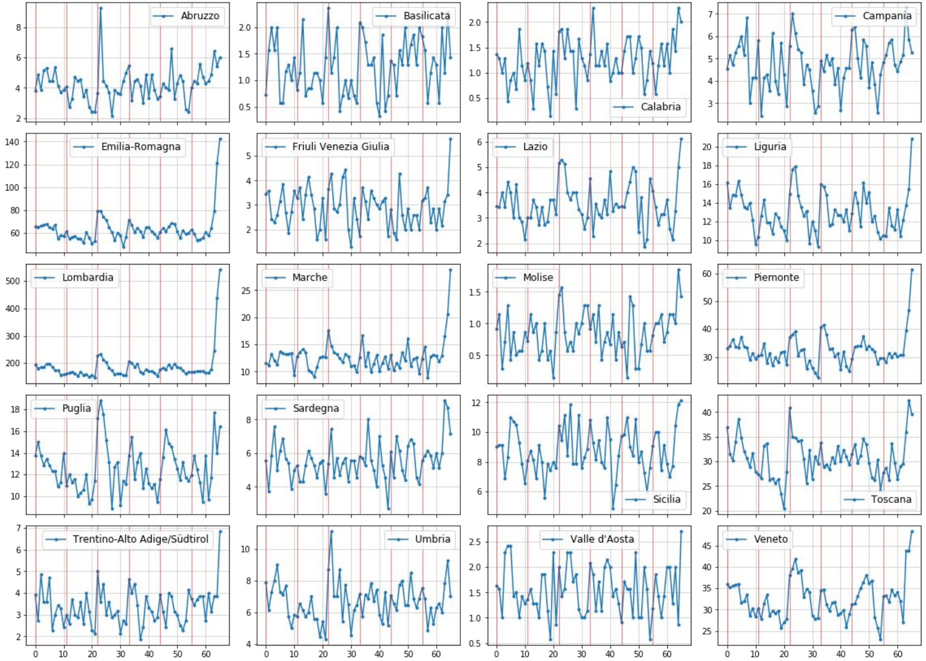
requires knowledge of two numbers:

1) total number of deaths
2) total number of infections

a lot of discussion has focused on getting 2) right (e.g. antibody tests)

Are we sure that all deaths have been counted? Can we find a dataset that determines 1) independently of testing?

daily mortality with age information for 1648 towns from all the different regions in Italy from 01/01-04/04 for 2015-2020



Counterfactual Analysis

What would the mortality have been had there been no pandemic?

Option 1: mean estimate and standard deviation on mean
completely agnostic of this year's data and correlations between weeks
↦ suboptimal and possibly biased.

Our Analysis:

1. learn a generative model from historic data
2. condition it on this year's pre-pandemic data
3. make counterfactual (what if there had been no pandemic) prediction for time after onset of pandemic (Feb 22)
4. compare factual to counterfactual to estimate the excess.
5. compare the excess to official COVID-19 death count

Conditional Mean Gaussian Process

Assumption: data follows a Gaussian distribution

Method: Train a GP and condition on this year's pre-pandemic data

Choice of kernel:

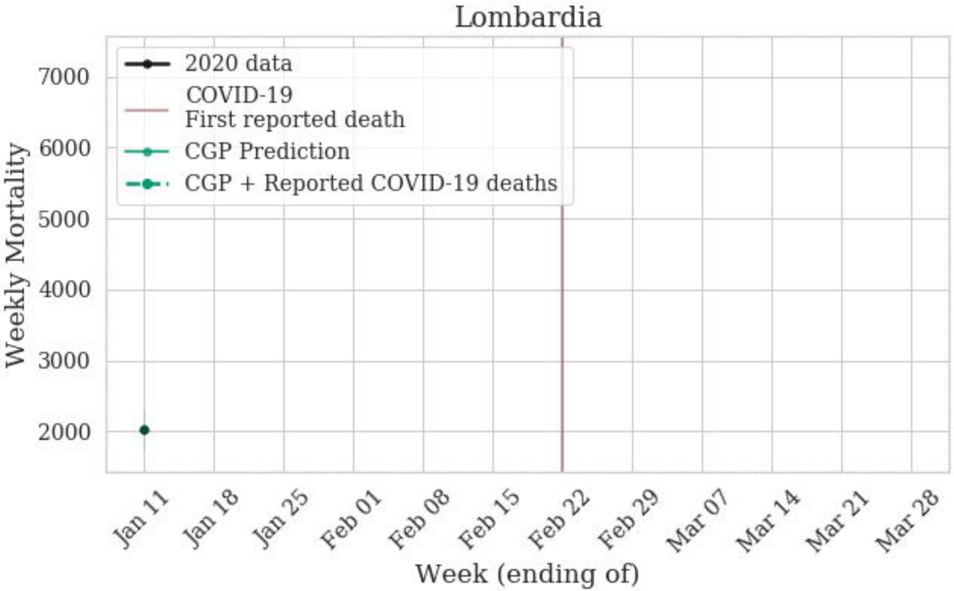
- learn kernel from historical data:
  - 2 component PCA of measured cov (captures 90% of variance in data)
  - add a squared exponential stationary kernel

Prediction is mean:

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}{}^{-1}(\boldsymbol{a} - \boldsymbol{\mu}_2)$$

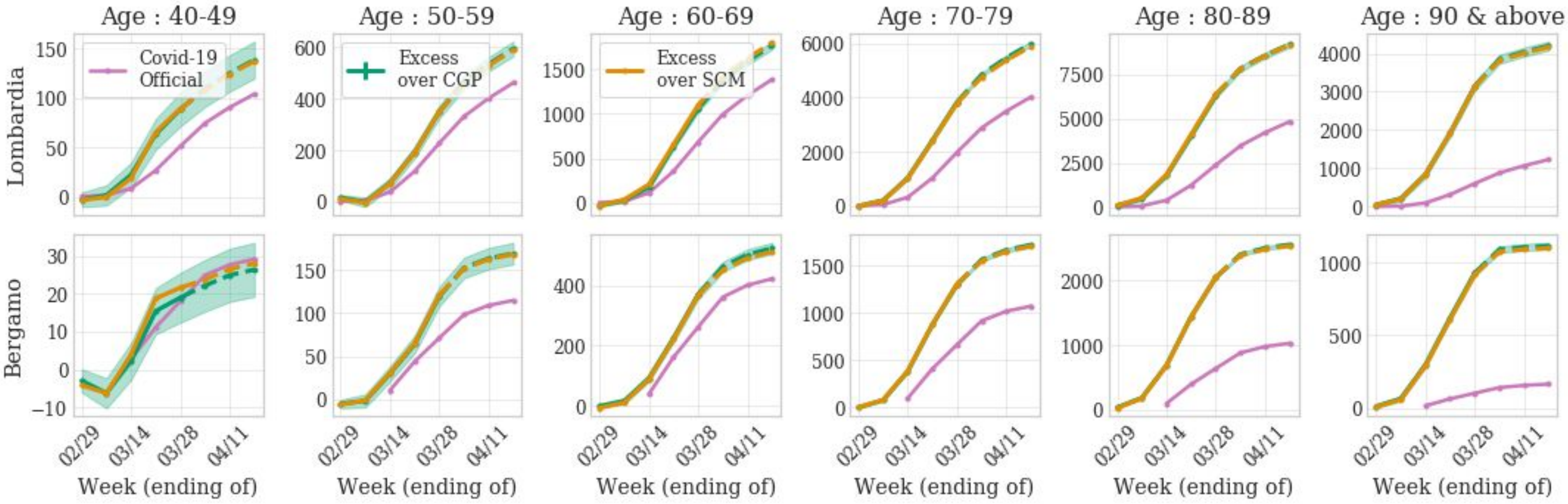
Uncertainty from covariance:

$$\overline{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}{}^{-1}\Sigma_{21}$$



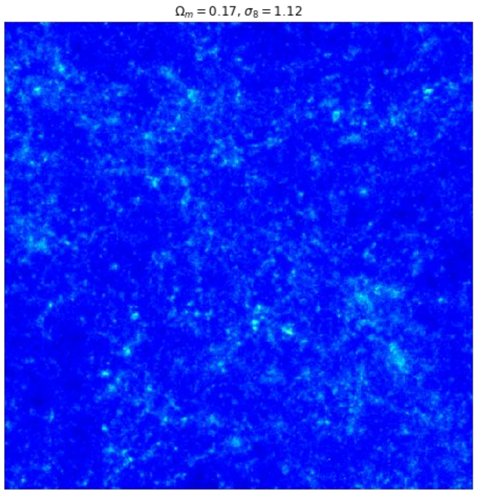
Cumulative Excess Mortality Count

- Fairly good match with reported COVID-19 deaths for ages below 50 years
- Significant excess that increases with age for ages above 60.
- *Hypothesis:* old people that die at home or in retirement homes; never tested for COVID-19



MADLens
a python package for fast, accurate, non-linear and differentiable lensing simulations

Vanessa Böhm, Yu Feng, Max Lee, Biwei Dai, in prep.



non linear lensing convergence field produced with the code

MADLens - fast, accurate, non-linear and differentiable lensing simulations

Motivation

- **Fast** allows for large simulation sets/on the fly simulations for data-hungry applications
  - Machine Learning
  - Sampling/Monte Carlo

- **Non-linear + accurate**
  - simulations are used to explore non-linear regime

- **Differentiable** with respect to initial conditions *and cosmological parameters*
  - Bayesian inverse problems
  - optimal data compression
  - coupling with ML methods

Max Lee

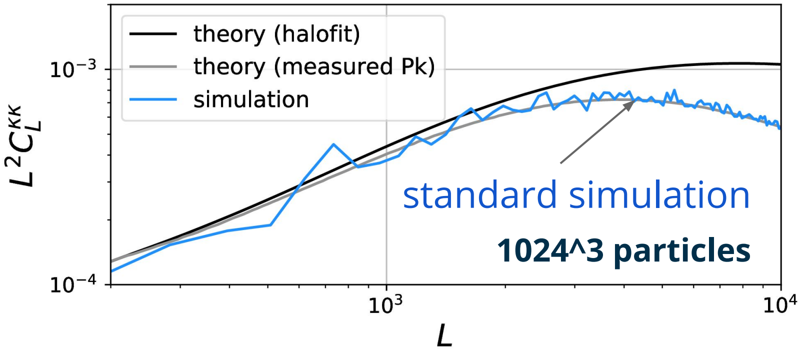
MADLens - what it does differently

Challenges

- degree-scale field of view
- arcmin resolution
- noiseless, accurate differentiation
- scalability (parallelized)

Our solutions

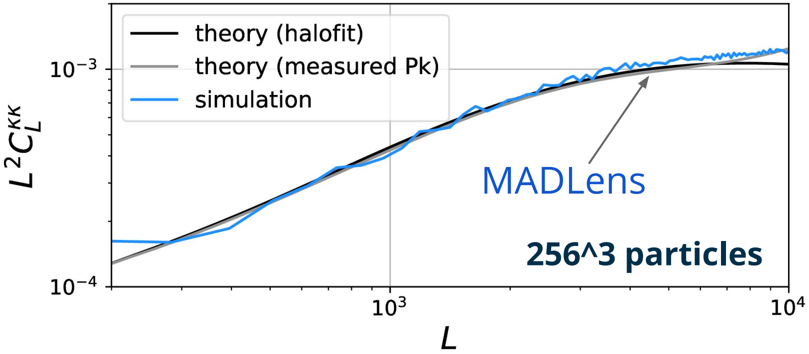
- scalable particle mesh framework (**pmesh, fastpm,** Feng et al. 1603.00476)
- ‘sharpening’ with Potential Gradient Descent (**PGD**, Dai et al. 1804.00671)
- **VMAD** (virtual machine automated differentiation framework; by Yu Feng)
- interpolation between snapshots



standard simulation

1024^3 particles

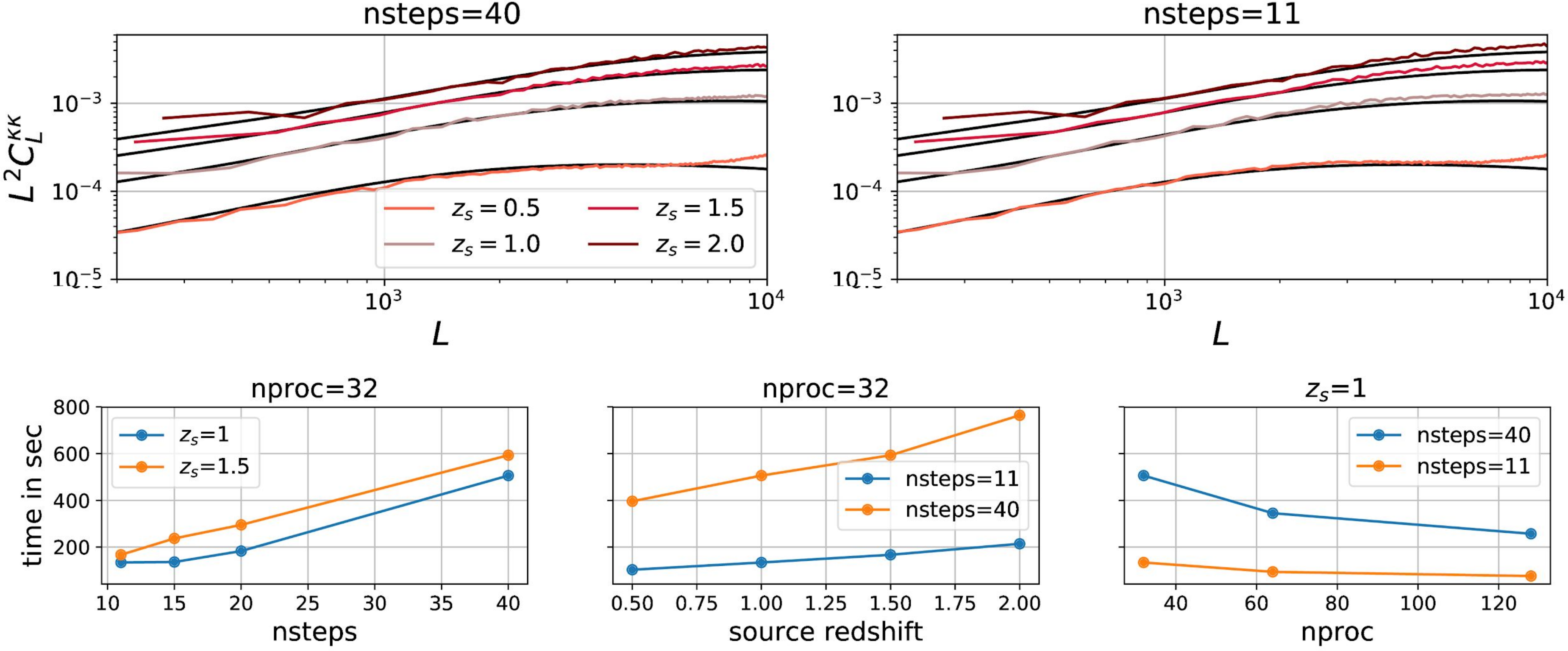
FOV 6.3 deg, 256 Mpc/h box, z_source =1



MADLens

256^3 particles

MADLens - benchmarks



The End

Questions?