Bayesian inference, calibration, optimization, sampling
(The MAD group presents)

Emil, Julie, Nesar, Ahmed, Todd + Yaohang, Kishan

Emil: Leave Fri around 6pm/not available on Sat
Julie: around all the time
Nesar: around Thr, Fri/may be remote Sat.
Ahmed: will be remote
Todd: will be remote Thr/potentially Fri

Oct 21, 2022

Topics Related to MAD

understand problem → abstraction/foundations → computations ↔ HPC (production)
problem methods

1. Define the problem at a high-level of abstract
2. Explore strategies to qualify (probabilistic model) and quantify uncertainties (evaluate) and a process to introduce them in the framework (computations)
3. Check for consistency: the framework is statistically consistent and establish validation procedures. Corollary: statistical testing
4. Dimensionality analysis - provide support for model reduction
5. Aspects regarding sampling: efficient MCMC like parallelization
6. Statistics for GANN-ML. What are the statistical assumptions and asymptotics
7. Result interpretation: Bayesian, frequentist

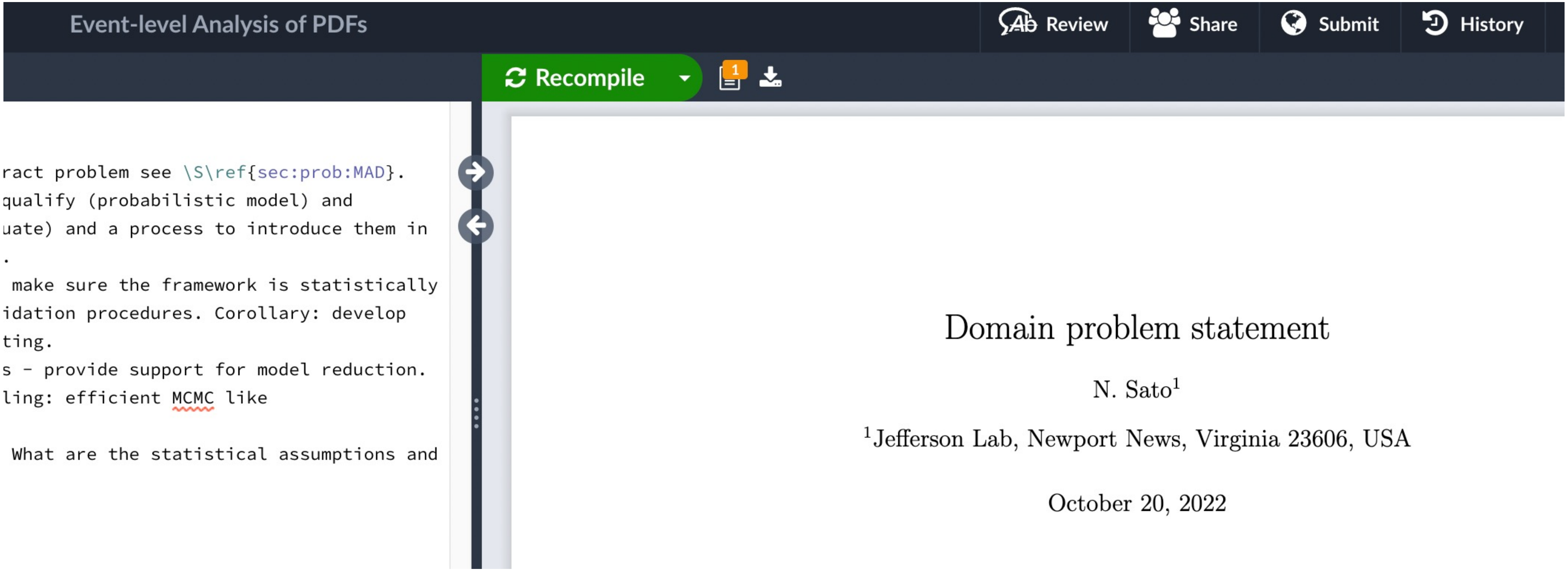
Outline

- Problem abstraction
- MCMC parallelization
- Model Reduction
- GAN

Addendum

- Scoring strategy

Problem Abstraction (0/6)



Problem Abstraction (1/6)

Suppose $f(\mathbf{p}_{\text{Thr}}) \leftarrow f_i(\xi, \mu_0^2)$ is defined on functional space $(f_i(\xi) = p_{i,0}\xi^{p_{i,1}}(1-\xi)^{p_{i,2}}(1+p_{i,3}\xi+\cdots)$.
We assume that we can evaluate quantity $\mathrm{d}\sigma \leftarrow \frac{d\sigma^{\text{NC}}(x,Q^2)}{dxdQ^2}$ pointwise:

$$\mathrm{d}\sigma(\mathbf{p}_{\text{Thr}}) = G(f(\mathbf{p}_{\text{Thr}})) \quad \text{or} \quad \frac{d\sigma^{\text{NC}}(x,Q^2)}{dxdQ^2} = G(f_i(\xi, \mu_0^2))\,.$$

We assume that we have events distributed as data $= \mathbf{d} \sim \mathrm{d}\sigma_{\text{true}}$

Also assume there is a detector model (unfolding) with parameters $\mathbf{p}_{\text{Det}}$: $\mathrm{d}\sigma_{\text{obs}} \sim D(\mathrm{d}\sigma_{\text{true}}; \mathbf{p}_{\text{Det}})$

Summary:
- Theory: $\mathrm{d}\sigma(\mathbf{p}_{\text{Thr}}) = G(f(\mathbf{p}_{\text{Thr}}))$
- Data: $\mathbf{d} \sim \mathrm{d}\sigma_{\text{true}}$
- Detector: $\mathrm{d}\sigma_{\text{obs}} \sim D(\mathrm{d}\sigma_{\text{true}}; \mathbf{p}_{\text{Det}})$
- Parameters $\mathbf{p} = [\mathbf{p}_{\text{Thr}}, \mathbf{p}_{\text{Det}}]$: (i) theory $\mathbf{p}_{\text{Thr}}$ and (ii) detector $\mathbf{p}_{\text{Det}}$

Problem Abstraction (2/6)

Mock of a model:

Data model: $\pi(\mathrm{d}\sigma | G_{\mathrm{Thr}}, \mathbf{p}_{\mathrm{Det}})$
Theory model: $\pi(G_{\mathrm{Thr}} | \mathbf{p}_{\mathrm{Thr}})$
Parameter model: $\pi(\mathbf{p})$

Joint distribution:

$$\pi(\mathrm{d}\sigma, G_{\mathrm{Thr}}, \mathbf{p}) = \pi(\mathrm{d}\sigma, G_{\mathrm{Thr}} | \mathbf{p}_{\mathrm{Det}}) \pi(G_{\mathrm{Thr}} | \mathbf{p}_{\mathrm{Thr}}) \pi(\mathbf{p})$$

Inference:

$$\pi(\mathbf{p}_{\mathrm{Thr}} | G_{\mathrm{Thr}}, \mathrm{d}\sigma, \mathbf{p}_{\mathrm{Det}}) = \pi(\mathrm{d}\sigma, G_{\mathrm{Thr}} | \mathbf{p}_{\mathrm{Det}}) \pi(G_{\mathrm{Thr}} | \mathbf{p}_{\mathrm{Thr}}) \pi(\mathbf{p})$$

Problem Abstraction (3/6)

Bayesian Approach to Inverse Problems: an Application to NNPDF Closure Testing

Luigi Del Debbio[a], Tommaso Giani[b,c], and Michael Wilson[a]

2.1 Statement of the problem

The space of inputs is denoted by X, while R denotes the space of responses. The model is specified by a *forward map*

$$\begin{aligned} G:\ & X \to R \\ & u \mapsto r = G(u)\,, \end{aligned} \tag{1}$$

Experiments will not have access to the full function r but only to a subset of N_{data} observations. In order to have a formal mathematical expression that takes into account the fact that we have a finite number of measurements, we introduce an *observation operator*

$$\begin{aligned} O:\ & R \to Y \\ & r \mapsto y\,, \end{aligned} \tag{3}$$

where $y \in Y$ is a vector in a finite-dimensional space Y of experimental results, *e.g.* the value of the structure function for some values of the kinematic variables x and Q^2. In general we will assume that $y \in \mathbb{R}^{N_{\text{data}}}$, *i.e.* we have a finite number N_{data} of real experimental values. The quantity of interest is the composed operator

Problem Abstraction (4/6)

Bayesian Approach to Inverse Problems: an Application to NNPDF Closure Testing

Luigi Del Debbio[a], Tommaso Giani[b,c], and Michael Wilson[a]

The quantity of interest is the composed operator

$$\mathcal{G}: \; X \to \mathbb{R}^{N_{\rm data}}$$
$$\mathcal{G} = O \circ G \,, \tag{4}$$

which maps the input u to the set of data. Taking into account the fact that experimental data are subject to noise, we can write

$$y = \mathcal{G}(u) + \eta \,, \tag{5}$$

where η is a random variable defined over $\mathbb{R}^{N_{\rm data}}$ with probability density $\rho(\eta)$. We will refer to η as the *observational noise*. In this setting, the inverse problem becomes finding u given y. It is often the case that inverse problems are ill-defined in the sense that the solution may not exist, may not be unique, or may be unstable under small variations of the problem.

In solving the inverse problem, we are going to adopt a Bayesian point of view, as summarised *e.g.* in Ref. [2]: our prior knowledge about u is encoded in a prior probability measure μ_X^0, where the suffix X indicates that the measure is defined in the space of models, and the suffix 0 refers to the fact that this is a prior distribution. We use Bayes' theorem to compute the posterior probability measure of u given the data y, which we denote as $\mu_X^{\mathcal{G}}$. When the probability measure can be described by a probability density, we denote the probability densities associated to μ_X^0 and $\mu_X^{\mathcal{G}}$, by π_X^0 and $\pi_X^{\mathcal{G}}$ respectively. Then, using Eq. (5), we can write the data likelihood, *i.e.* the probability density of y given u,

$$\pi_Y(y|u) = \rho(y - \mathcal{G}(u)) \,, \tag{6}$$

and Bayes' theorem yields

$$\pi_X^{\mathcal{G}}(u) = \pi_X(u|y) \propto \pi_X^0(u)\rho(y - \mathcal{G}(u)) \,. \tag{7}$$

Problem Abstraction (5/6)

Bayesian Approach to Inverse Problems: an Application to NNPDF Closure Testing

Luigi Del Debbio[a], Tommaso Giani[b,c], and Michael Wilson[a]

altogether. The net effect of the theory errors is a redefinition of the covariance of the data, which has no major impact in our discussion, and therefore will be ignored. Taking the correlation $\theta(y, u|\mathcal{G})$ into account, the joint distribution of y and u is

$$\pi^{\mathcal{G}}(y, u|y_0, C_Y, u_0, C_X) \propto \pi_X^0(u|u_0, C_X)\pi_Y^0(y|y_0, C_Y)\theta(y, u|\mathcal{G}) . \tag{14}$$

We can now marginalize with respect to y, neglecting theory errors,

$$\pi_X^{\mathcal{G}}(u|y_0, C_Y, u_0, C_X) \propto \int dy\, \pi_X^0(u|u_0, C_X)\pi_Y^0(y|y_0, C_Y)\theta(y, u|\mathcal{G}) \tag{15}$$

$$\propto \pi_X^0(u|u_0, C_X) \int dy\, \pi_Y^0(y|y_0, C_Y)\delta\left(y - \mathcal{G}(u)\right) \tag{16}$$

$$\propto \pi_X^0(u|u_0, C_X)\, \pi_Y^0(\mathcal{G}(u)|y_0, C_Y) . \tag{17}$$

We see that we have recovered Eq. 7. The log-likelihood in the Gaussian case is simply the χ^2 of the data, y_0, to the theory prediction, $\mathcal{G}(u)$:

$$-\log \pi_Y^0(\mathcal{G}(u)|y_0, C_Y) = \frac{1}{2}\sum_{i,j=1}^{N_{\rm data}} \left(\mathcal{G}(u) - y_0\right)_i \left(C_Y^{-1}\right)_{ij} \left(\mathcal{G}(u) - y_0\right)_j . \tag{18}$$

In the notation of Eq. 7

$$\pi_Y^0(\mathcal{G}(u)|y_0, C_Y) = \rho\left(\mathcal{G}(u) - y_0\right) , \tag{19}$$

where in this case

$$\rho(\eta) \propto \exp\left(-\frac{1}{2}\left|\eta\right|_{C_Y}^2\right) . \tag{20}$$

Problem Abstraction (6/6)

Luigi Del Debbio[a], Tommaso Giani[b,c], and Michael Wilson[a]

that becomes increasingly difficult in high-dimensional spaces. As discussed later in this study, the NNPDF approach is focused on the determination of the *Maximum A Posteriori (MAP)* estimator, *i.e.* the element $u_* \in X$ that maximises $\pi_X^{\mathcal{G}}(u)$:

$$u_* = \arg\min_{u \in X} \left(\frac{1}{2} \left| y_0 - \mathcal{G}(u) \right|_{C_Y}^2 + \frac{1}{2} \left| u - u_0 \right|_{C_X}^2 \right) . \tag{23}$$

For every instance of the data y_0, the MAP estimator is computed by minimising a regulated χ^2, where the regularization is determined by the prior that is assumed on the model u. We will refer to this procedure as the *classical fit* of experimental data to a model. Note that in the Bayesian approach, the regulator appears naturally after having specified carefully all the assumptions that enter in the prior. In this specific example the regulator arises from the Gaussian prior for the model input u, which is normally distributed around a solution u_0. The MAP estimator provides the explicit connection between the Bayesian approach and the classical fit.

Problem Abstraction (-1/6)

This cites the above.

The Path to Proton Structure at One-Percent Accuracy

The NNPDF Collaboration:

Richard D. Ball,[1] Stefano Carrazza,[2] Juan Cruz-Martinez,[2] Luigi Del Debbio,[1] Stefano Forte,[2] Tommaso Giani,[1,8] Shayan Iranipour,[3] Zahari Kassabov,[3] Jose I. Latorre,[4,5,6] Emanuele R. Nocera,[1,8] Rosalyn L. Pearson,[1] Juan Rojo,[7,8] Roy Stegeman,[2] Christopher Schwan,[2] Maria Ubiali,[3] Cameron Voisey,[9] and Michael Wilson[1]

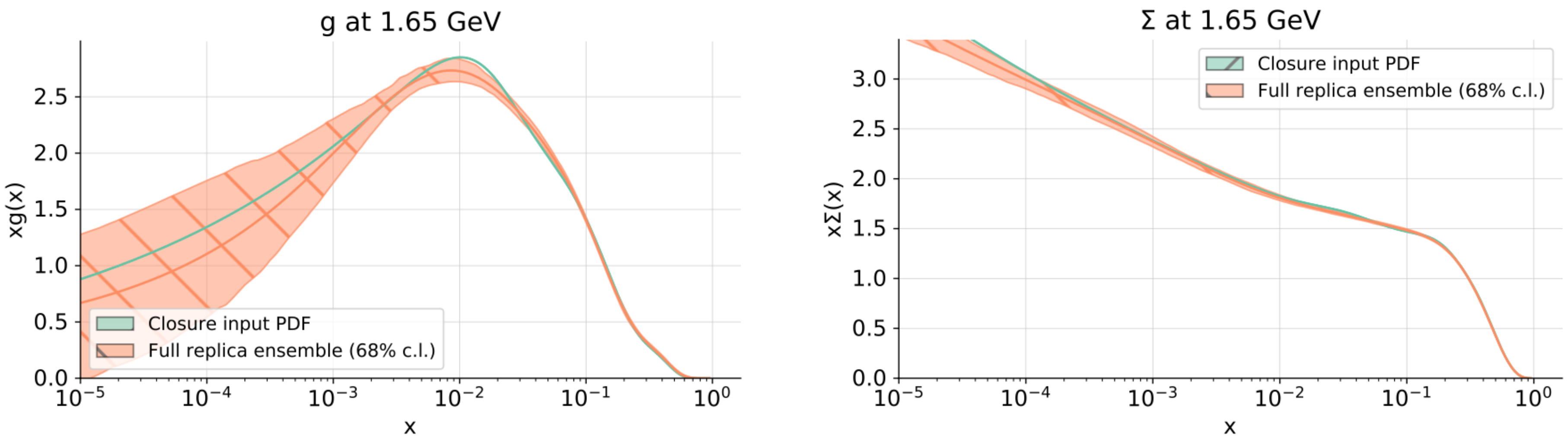
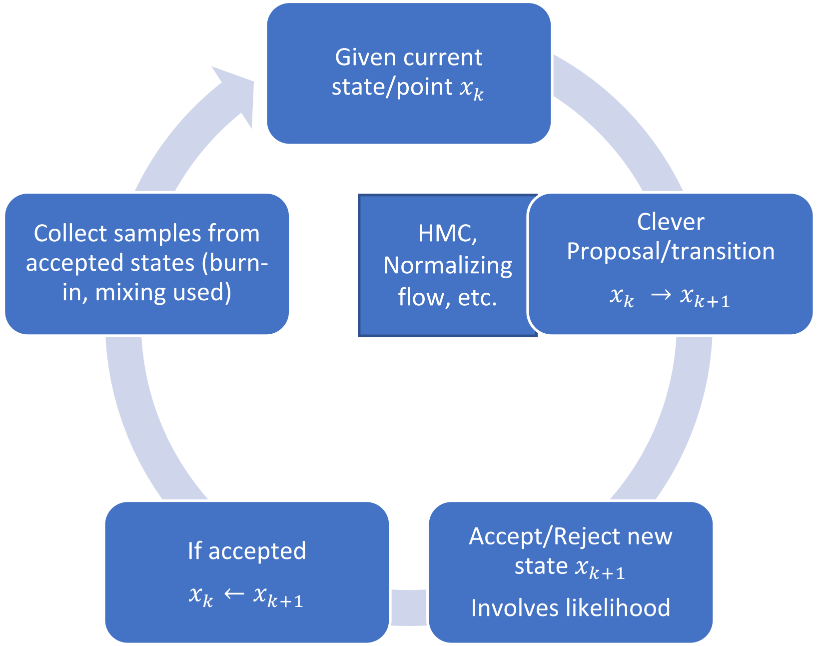


Figure 6.1. The replica (solid green line) chosen as the true underlying PDF $\boldsymbol{f}$ for the closure test: the gluon (left) and quark singlet (right) are displayed. The NNPDF4.0 central value and 68% confidence interval (same as in Fig. 5.2) are also shown for reference.

Aspects regarding sampling: efficient MCMC like parallelization (1/4)



Facts:

- MCMC is inherently serial/sequential due to the Markov property of the chain
- Parallelizing MCMC such as to guarantee convergence to the exact posterior *is not easy*
- ***Practical approaches*** for ***parallelizing MCMC*** with asymptotic guarantees ***do exist***, but there is ***no absolute winner***

Target distribution (posterior):

- MCMC seeks to converge and collect samples from a distribution (usually posterior):

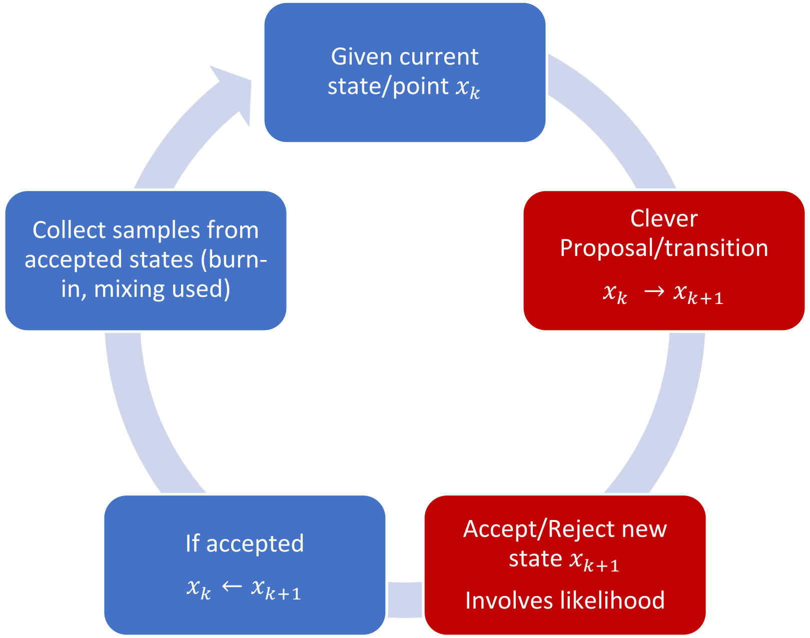
$$\pi(x) \propto p(y|x)\, p(x)$$

Target (unnormalized) posterior

Data-likelihood (expensive) (simulation, experiment, etc.)

Prior inexpensive (once defined!)

Aspects regarding sampling: efficient MCMC like parallelization (2/4)



Parallelizing MCMC:

1. **Single chain:** parallelize the likelihood (e.g., simulation, proposal) and/or multi proposal
   - ***Pros**:* asymptotic convergence is guaranteed
   - ***Cons**:* parallelization gain is not significant

2. **Multiple (multi) chains:** run multiple chains (with same proposal) in parallel:
   - ***Pros**:* considerable speedup; multilevel (chains & likelihood) parallelization
   - ***challenges**:* cost of burn-in on each chain; asymptotic convergence is possible but requires clever convergence diagnostics and careful aggregation of the parallel samples collected;
   - **General Approaches**: (each chain sample on its own, then samples are ***aggregated!***
     - Parallel chains (same data & kernel/proposal, but different random seeds/sequences) explore the whole domain; parallel tampering, equi-energy, etc. can be useful but still costly
     - Data and/or parameter space splitting: one chain runs per sub-set/domain
     - Approximations using normalizing flows
     - Kernel approximation: e.g., GMM with parallel chains per kernel, region, sub-set/domain

Aspects regarding sampling: efficient MCMC like parallelization (3/4)

<u>So, which MCMC parallelization approach?</u>

- *As mentioned earlier, there is no absolute winner!*
- We should build a battery of parallel MCMC implementations and test their performance (both accuracy and computational cost), optimally tune each implementation, and explore possibility of hybridization
- We can implement and test these samplers fairly quickly with a proxy or realistic toy model; the winner(s) can then be implemented in the full setup

Aspects regarding sampling: efficient MCMC like parallelization (4/4)

References[Partial list]:

1. Glatt-Holtz, Nathan E., Andrew J. Holbrook, Justin A. Krometis, and Cecilia F. Mondaini. "Parallel MCMC Algorithms: Theoretical Foundations, Algorithm Design, Case Studies." *arXiv preprint arXiv:2209.04750* (2022)
2. Robert, Christian P., Víctor Elvira, Nick Tawn, and Changye Wu. "Accelerating MCMC algorithms." *Wiley Interdisciplinary Reviews: Computational Statistics* 10, no. 5 (2018): e1435.
3. Albergo, Michael S., Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. "Introduction to normalizing flows for lattice field theory." arXiv preprint arXiv:2101.08176 (2021).
4. Hafych, Vasyl, Philipp Eller, Oliver Schulz, and Allen Caldwel. "Parallelizing mcmc sampling via space partitioning." Statistics and Computing 32, no. 4 (2022): 1-14.
5. Wang, Xiangyu, Fangjian Guo, Katherine A. Heller, and David B. Dunson. "Parallelizing MCMC with random partition trees." Advances in neural information processing systems 28 (2015).
6. Neiswanger, Willie, Chong Wang, and Eric Xing. "Asymptotically exact, embarrassingly parallel MCMC." arXiv preprint arXiv:1311.4780 (2013).
7. Calderhead, Ben. "A general construction for parallelizing Metropolis− Hastings algorithms." *Proceedings of the National Academy of Sciences* 111, no. 49 (2014): 17408-17413.
8. De Souza, Daniel A., Diego Mesquita, Samuel Kaski, and Luigi Acerbi. "Parallel MCMC Without Embarrassing Failures." In *International Conference on Artificial Intelligence and Statistics*, pp. 1786-1804. PMLR, 2022.
9. Attia, Ahmed, Azam Moosavi, and Adrian Sandu. "Cluster sampling filters for non-Gaussian data assimilation." *Atmosphere* 9, no. 6 (2018): 213.

Dimensionality Analysis and Model Reduction

Proxy model

Parton distribution functions

u(x) = cu*x**au*(1-x)**bu
d(x) = cd*x**ad*(1-x)**bd
cu, au, bu, cd, ad, bd: parameters of interest

Cross-sections (experimentally inferred):

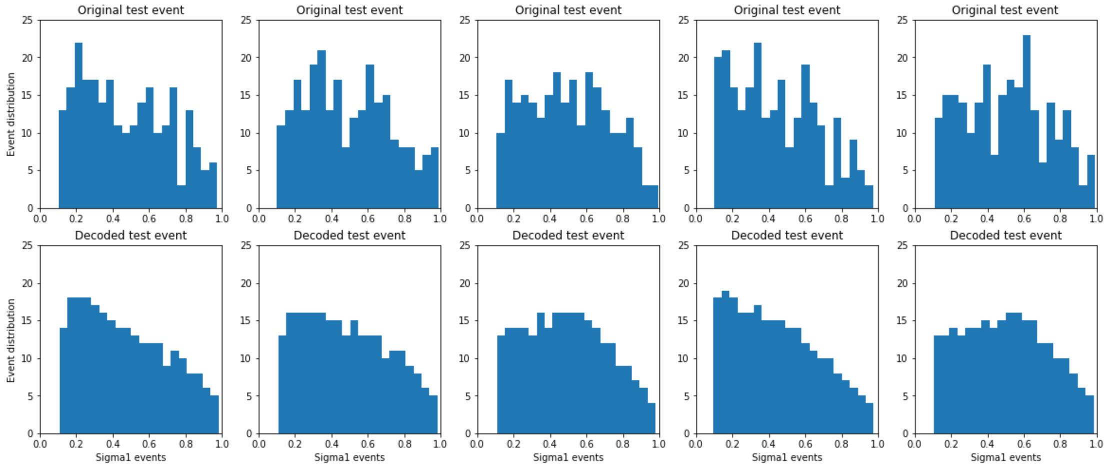
Sigma1(x) = 4u(x)+ d(x)
Sigma2(x) = 4d(x)+ u(x)

Exploration underlying latent space (event space):

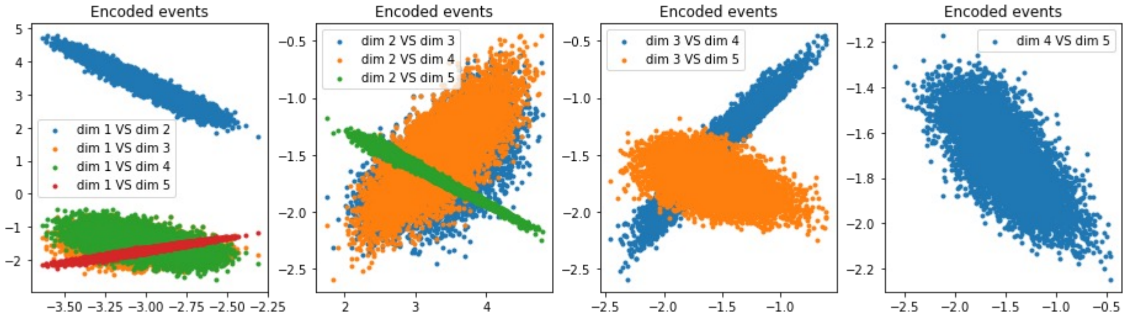
Auto-encoder: 4-layer encoder and 4-layer decoder

events ~ AE(events)

Decoded events

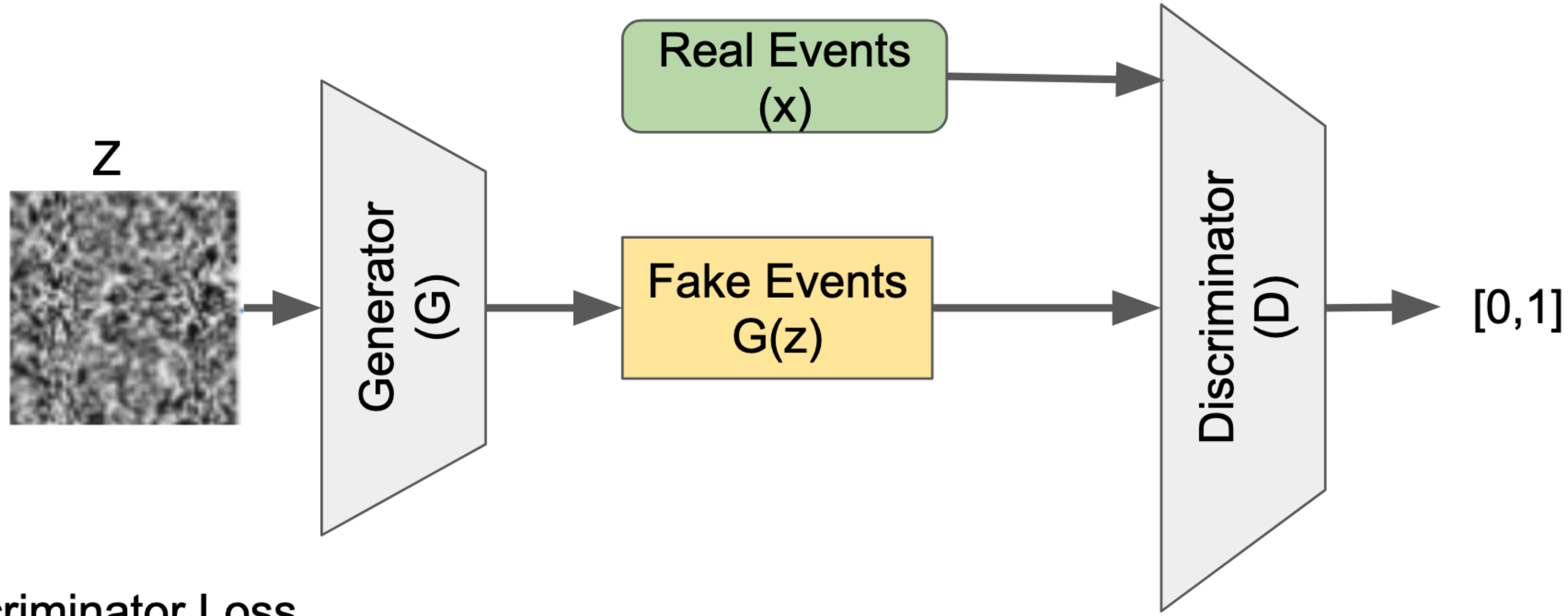


Latent space: encoded test events



Future works: find adequate latent dimension and propose reduced parameterization

How GAN works?



Discriminator Loss

$$L^{(D)} = \max[log(D(x)) + log(1 - D(G(z)))]$$

Generator Loss

$$L^{(G)} = \min[log(D(x)) + log(1 - D(G(z)))]$$

$$L = \min_G \max_D [log(D(x)) + log(1 - D(G(z)))]$$

$$\min_G \max_D V(D,G) = \min_G \max_D \left(E_{x \sim P_{data}(x)}[logD(x)] + E_{z \sim P_z(z)}[log(1 - D(G(z)))]\right)$$

How GAN works?

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (1)$$

We will show in section 4.1 that this minimax game has a global optimum for $p_g = p_{\text{data}}$. We will then show in section 4.2 that Algorithm 1 optimizes Eq 1, thus obtaining the desired result.

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

$$C(G) = \max_D V(G, D)$$

Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$\mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log D^*_G(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D^*_G(\boldsymbol{x}))]$$

then p_g converges to p_{data}

Other Items & Things that MAD Likes

1. Postdoc position ANL-ASCR

2. MAD would like to keep the proxy model and keep updating it; although it will not have all theory components it should encapsulate abstractions (e.g., the current model)

3. Laptop-sized framework in the works:
    /dev in our github – Kishan will push soon
    Use ClassRegistry to manage components (plug stuff in/out)
    Promotes code consistency, helps unit testing

Addendum

Scoring Strategy (for the proxy problem) 1/5

Likelihood:

$$S_k(\sigma_{th}, \sigma_{obs}^{\{k\}}) = \mathbf{E}_P\|\sigma_{th} - \sigma_{obs}^{\{k\}}\| - \frac{1}{2}\mathbf{E}_P\|\sigma_{th}^{\{a\}} - \sigma_{th}^{\{b\}}\| \qquad \forall \sigma_{th}^{a}, \sigma_{th}^{\{b\}} \sim P = \mathrm{Prob}(\mathbf{p})$$

$$= \frac{1}{N_s}\sum_{i=1}^{N_s} ||\sigma_{th} - \sigma_{obs}^{\{k\}}|| - \frac{1}{2{N_s}^2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} ||\sigma_{th}^{\{a\}} - \sigma_{th}^{\{b\}}||$$

$$S(\sigma_{th}, \sigma_{obs}) = \frac{1}{M}\sum_{k=1}^{M} S_k(\sigma_{th}, \sigma_{obs}^{\{k\}})$$

Loss function:

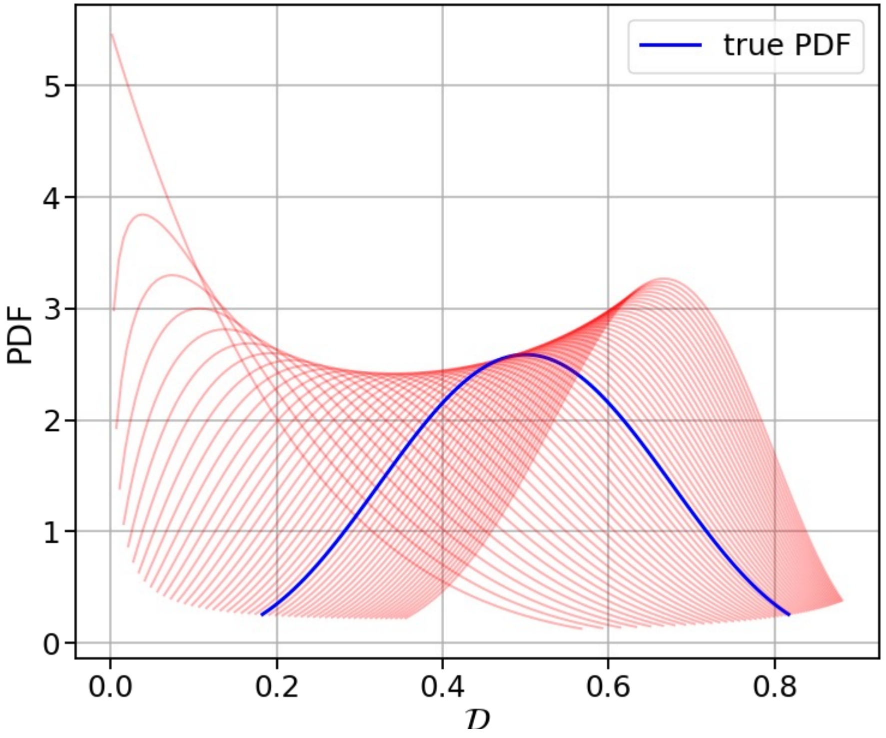
$$\mathcal{L} = \alpha_1 S(\sigma_{th}^{p}(\mathbf{p}), \sigma_{obs}^{p}) + \alpha_2 S(\sigma_{th}^{n}(\mathbf{p}), \sigma_{obs}^{n}) + \alpha_3 ||N_{obs}^{p} - N_{th}^{p}(\mathbf{p})|| + \alpha_4 ||N_{obs}^{n} - N_{th}^{n}(\mathbf{p})||$$

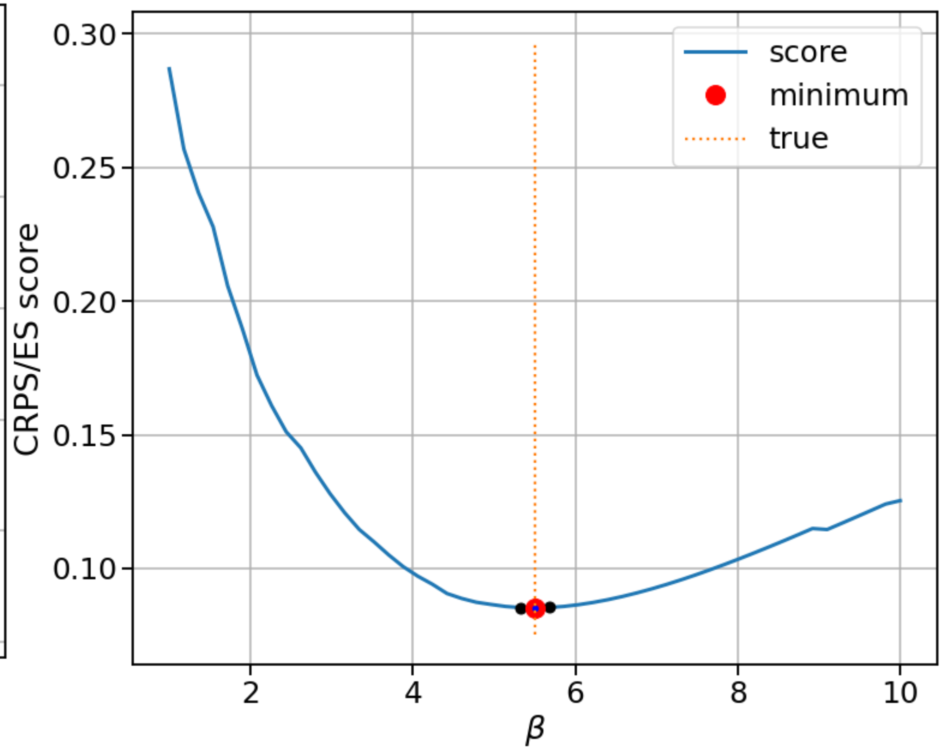
Remember GAN: $$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \qquad (1)$$

Scoring Strategy (for the proxy problem) 2/5

Toy-Example

Assume $X, Y \sim \text{Beta}(5.5, \mathbf{p})$. Given $M = 200$ samples from $X \sim \text{Beta}(5.5, \mathbf{p} = 5.5)$, find $\mathbf{p}$ by using $N_s = 1000$ samples from $Y \sim \text{Beta}(5.5, \mathbf{p})$.

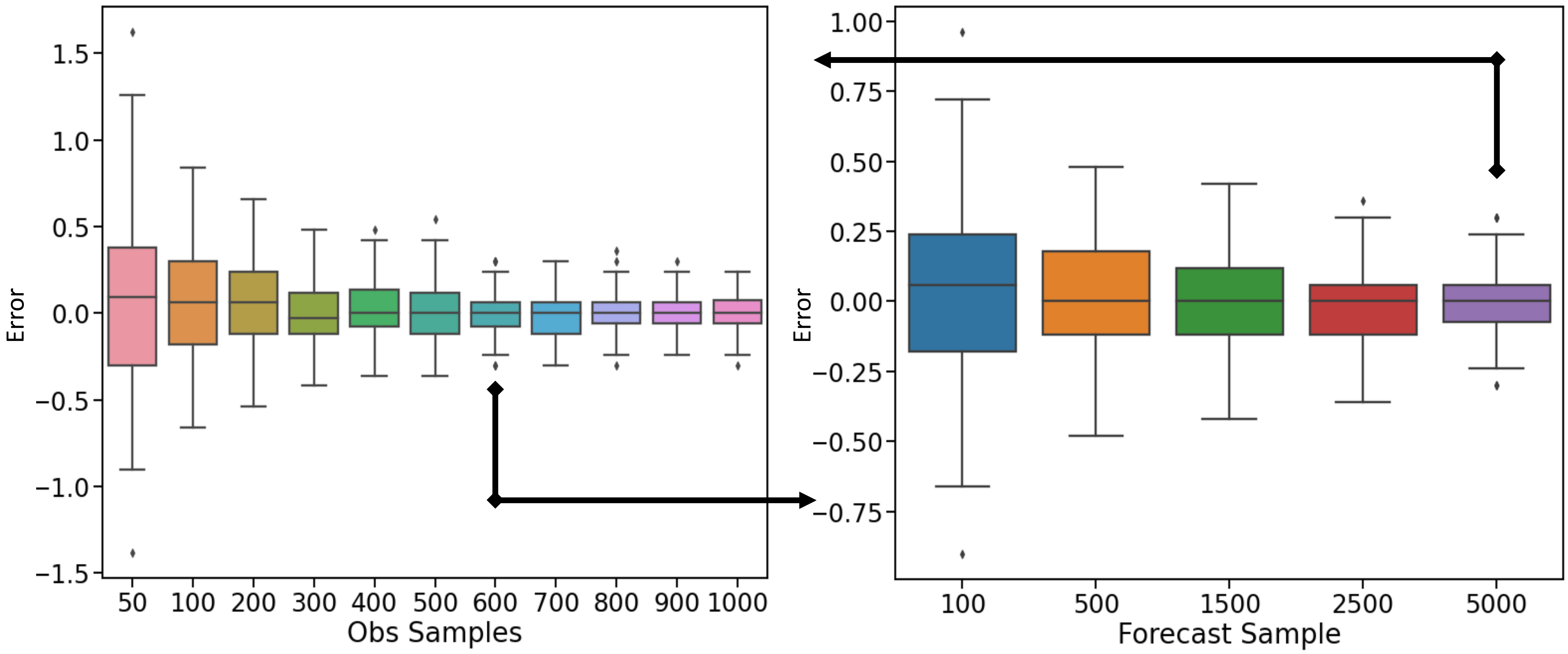




Scoring Strategy (for the proxy problem) 3/5

Convergence properties

$$\text{Error} = \mathbf{p}_{true} - \mathbf{p}^*, \text{ where } \mathbf{p}^* = \arg\min_{\mathbf{p}} \mathcal{L}(\mathbf{p})$$



Scoring Strategy (for the proxy problem) 4/5

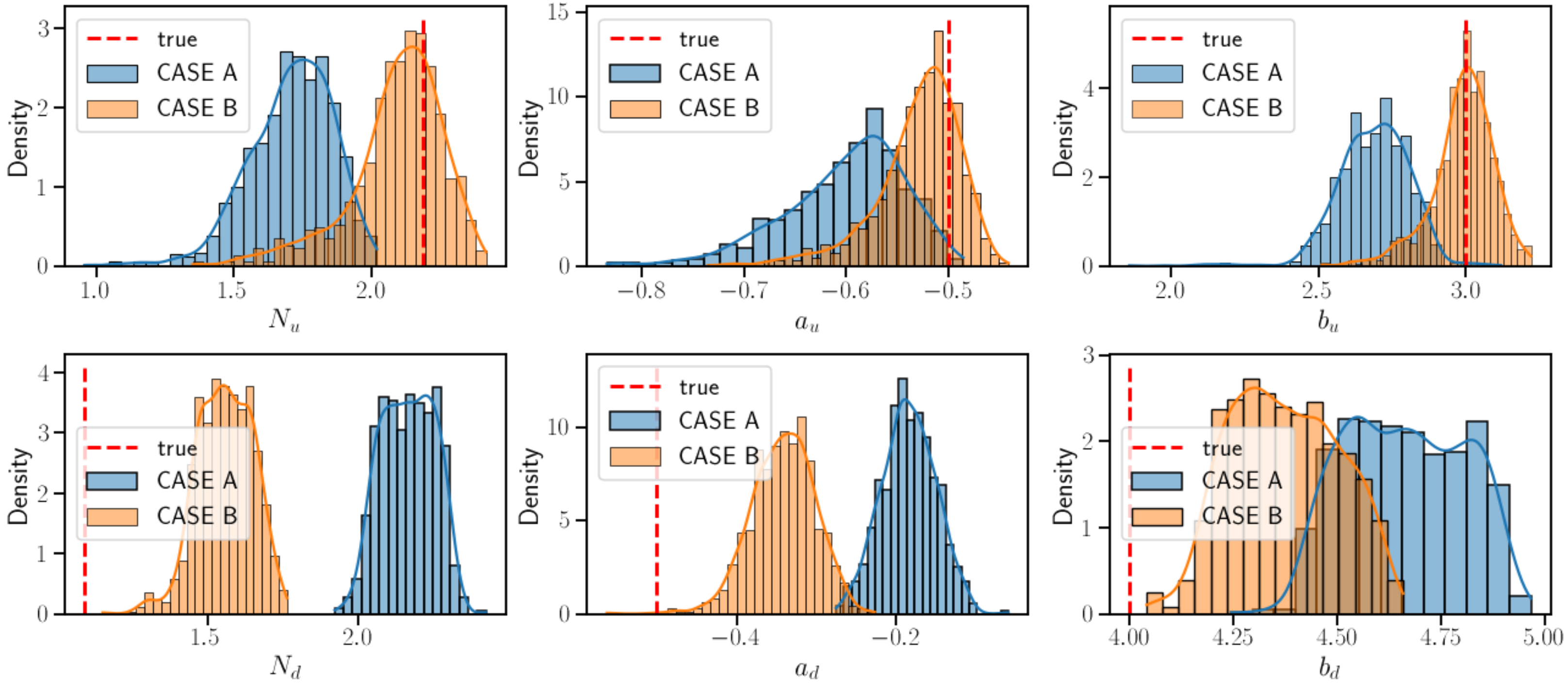
- 6-parameter model

$$u(x) = p_1 x^{p_2}(1-x)^{p_3}$$
$$d(x) = p_4 x^{p_5}(1-x)^{p_6}$$

$$d\sigma_u(x) = 4u(x) + d(x)$$
$$d\sigma_d(x) = u(x) + 4d(x)$$

- Data A: 1,000 events σ_1 ; 500 events σ_2
- Data B: 10,000 events σ_1 ; 5,000 events σ_2
- Data C: 100,000 events σ_1 ; 50,000 events σ_2

Nonparametric bootstrap distribution (obtained by sampling with replacement) gives direct CI estimates and approximates the posterior of a Bayesian problem with a specific un-informative prior.



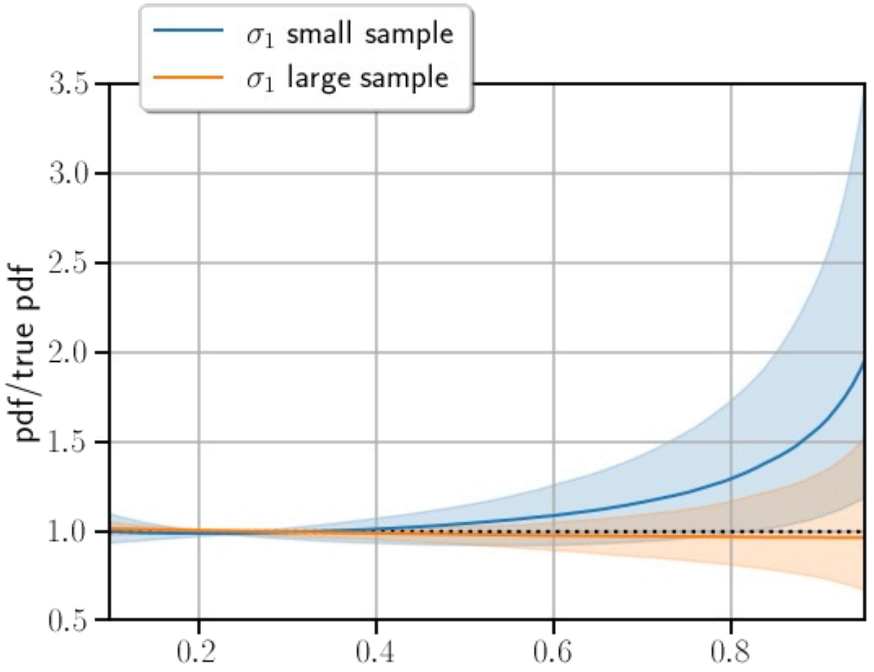
Scoring Strategy (for the proxy problem) 5/5

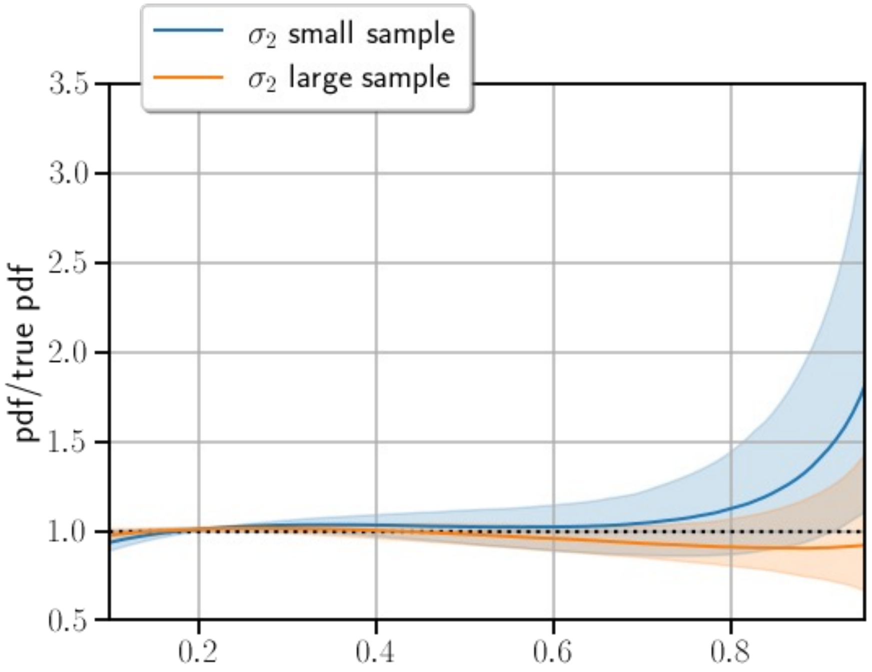
- 6-parameter model

$$u(x) = p_1 x^{p_2}(1-x)^{p_3} \qquad d\sigma_u(x) = 4u(x) + d(x)$$

$$d(x) = p_4 x^{p_5}(1-x)^{p_6} \qquad d\sigma_d(x) = u(x) + 4d(x)$$

- Data A: 1,000 events σ_1 ; 500 events σ_2
- Data B: 10,000 events σ_1 ; 5,000 events σ_2
- Data C: 100,000 events σ_1 ; 50,000 events σ_2





Auto-encoder for cross-section events: January 2022

Parton distribution functions:

- u(x) = cu*x**au*(1-x)**bu
- d(x) = cd*x**ad*(1-x)**bd
- s(x) = cs*x**as*(1-x)**bs

cu, au, bu, cd, ad, bd, cs, as, bs are 9 free parameters, drawn a priori from uniform distributions on [0, 1]

Cross-sections:

- Sigma1(x) = 4u(x)+ d(x)+ s(x)
- Sigma2(x) = 4d(x)+ u(x)+ s(x)
- Sigma3(x) = u(x)+ d(x)+ s(x)

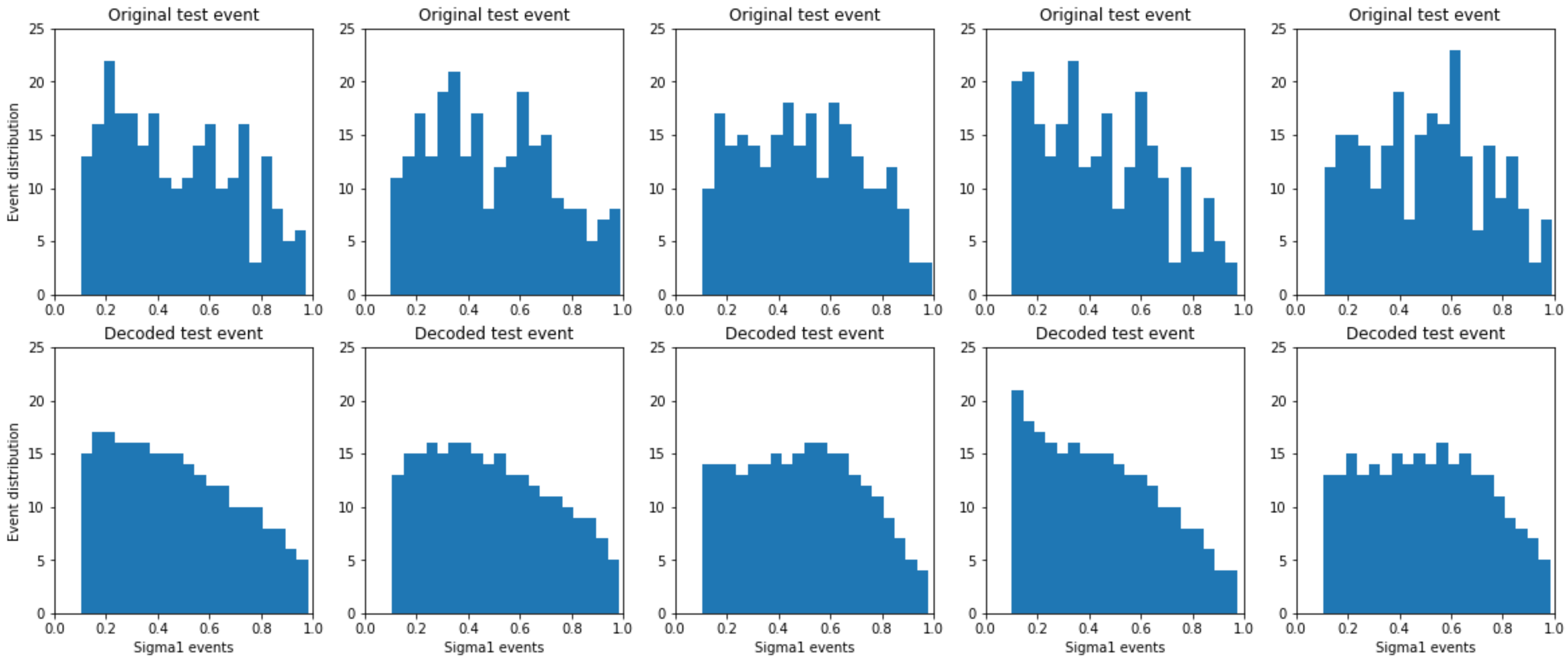
- 4-layer encoder (neurons per layer (activation): 250 (Relu), 64 (Relu), 32 (Relu), latent-space dim (Linear))
- 4-layer decoder (neurons per layer: latent-space dim (Relu), 32 (Relu), 64 (Relu), 250 (Sigmoid))
- events ~ AE(events)

- loss-function: binary-cross entropy
- 25 -> 45 epochs
- For each set of 9 parameters, 50 events are generated from each cross-section (sigma1, sigma2, sigma3)

Each cross-section generates 250 points per event

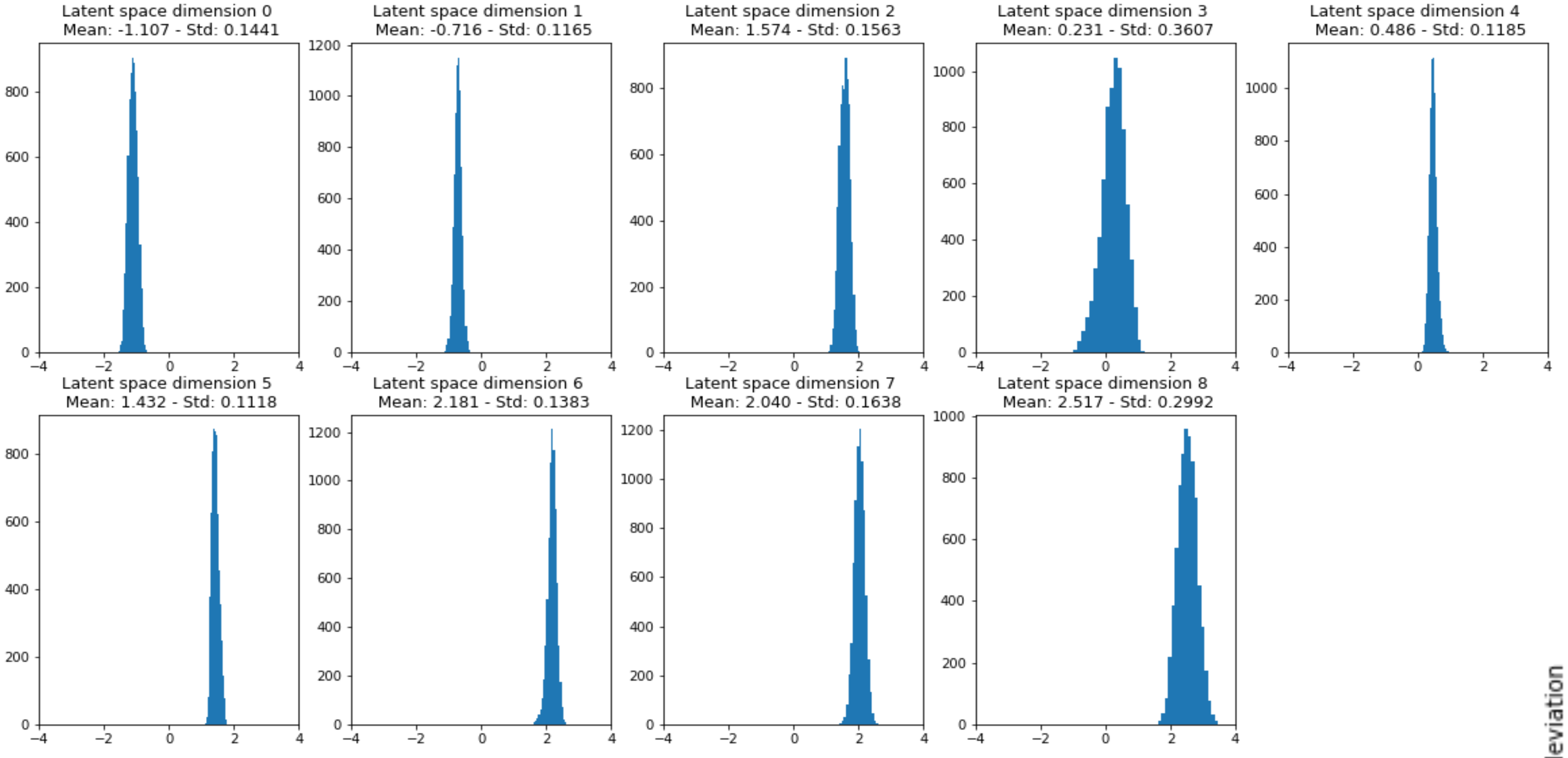
Julie Bessac

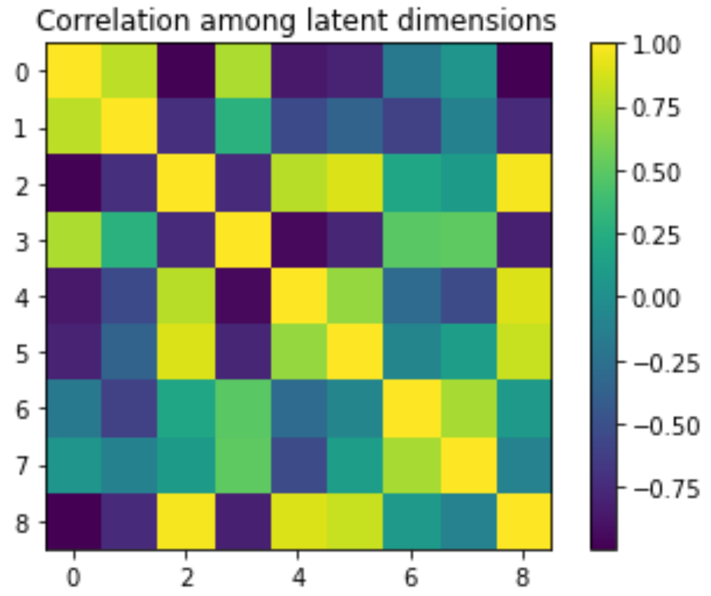
Latent space of dimension 9 – Encoded-decoded events

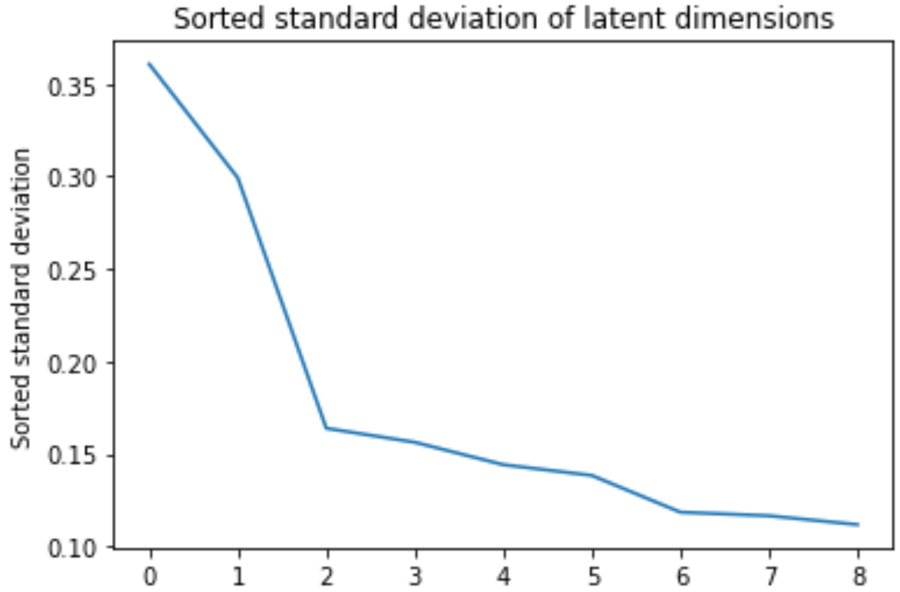


Julie Bessac

Latent space of dimension 9 – Latent space analysis



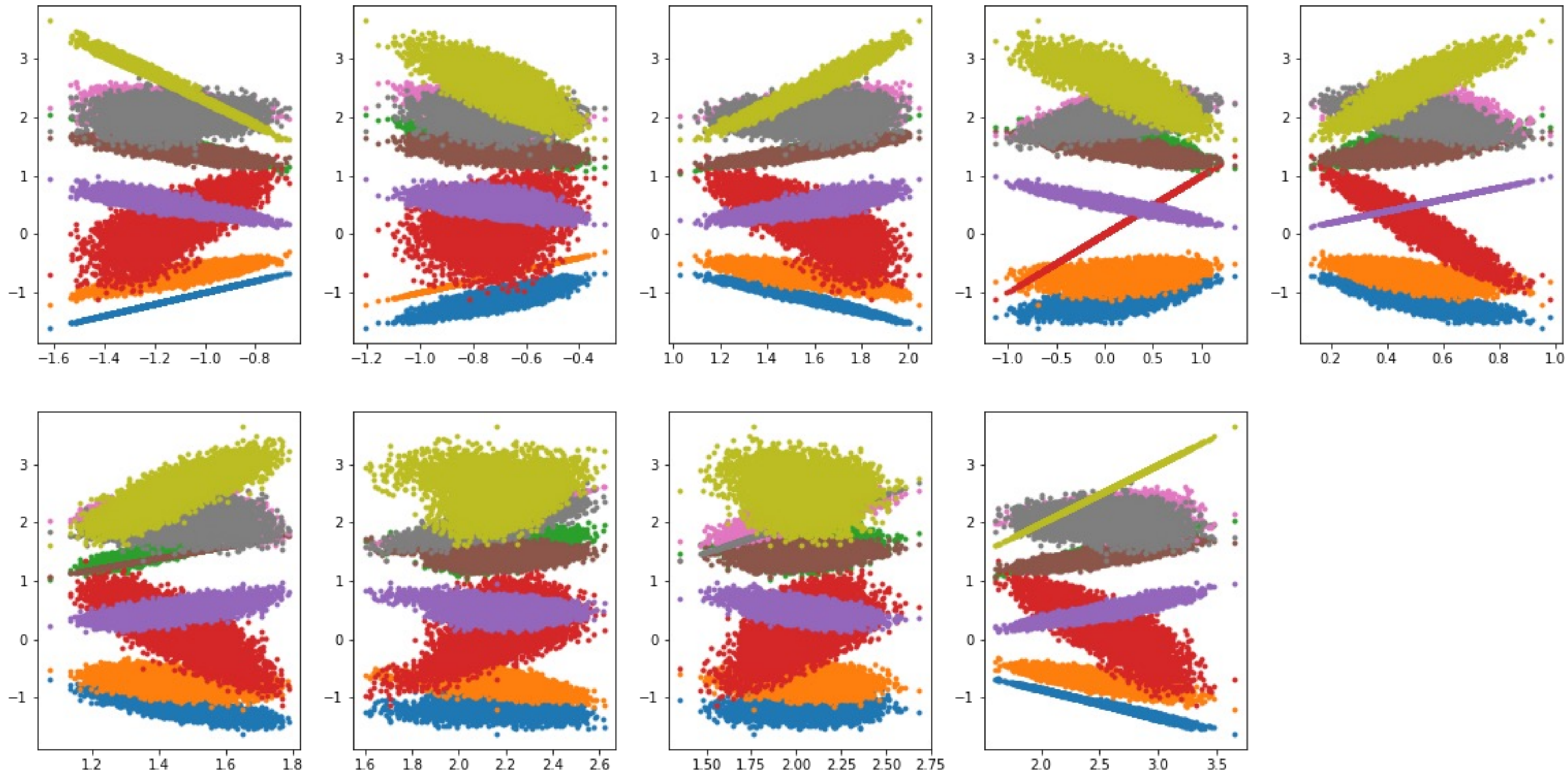




Julie Bessac

Latent space of dimension 9 – latent space analysis

Scatterplot of latent dimensions



Julie Bessac

Comparison of loss function for different AE

Dim 5 : loss: 0.5620 - val_loss: 0.5609
Dim 7 : loss: 0.5620 - val_loss: 0.5609
Dim 9 : loss: 0.5620 - val_loss: 0.5608

What's next:

- Shall we learn a parameterization of the reduced space?

- How to select the optimal latent dimension?

Julie Bessac