Blending Physics-Based numerical simulations and seismic databases using Generative Adversarial Networks (GAN)

D. Clouteau, F. Gatti

Department of Mechanics and Civil Engineering, LMSSMat UMR 8579
CentraleSupelec, France

Thanks to the SINAPS and SEISM Institute Team and in particular Régis Cottereau, Fernando Lopez-Caballero

October 3, 2019

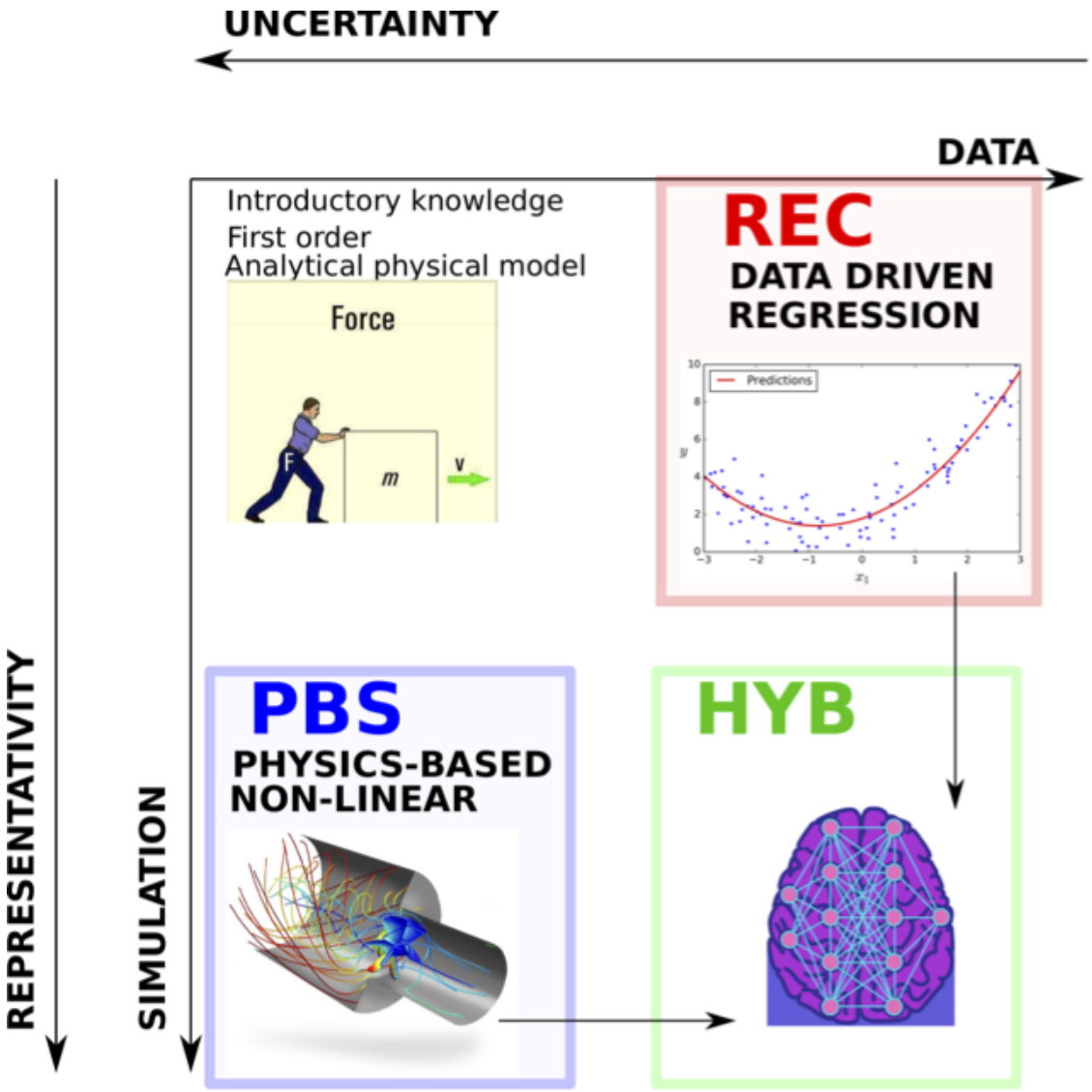
Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Physics-based or Data-based



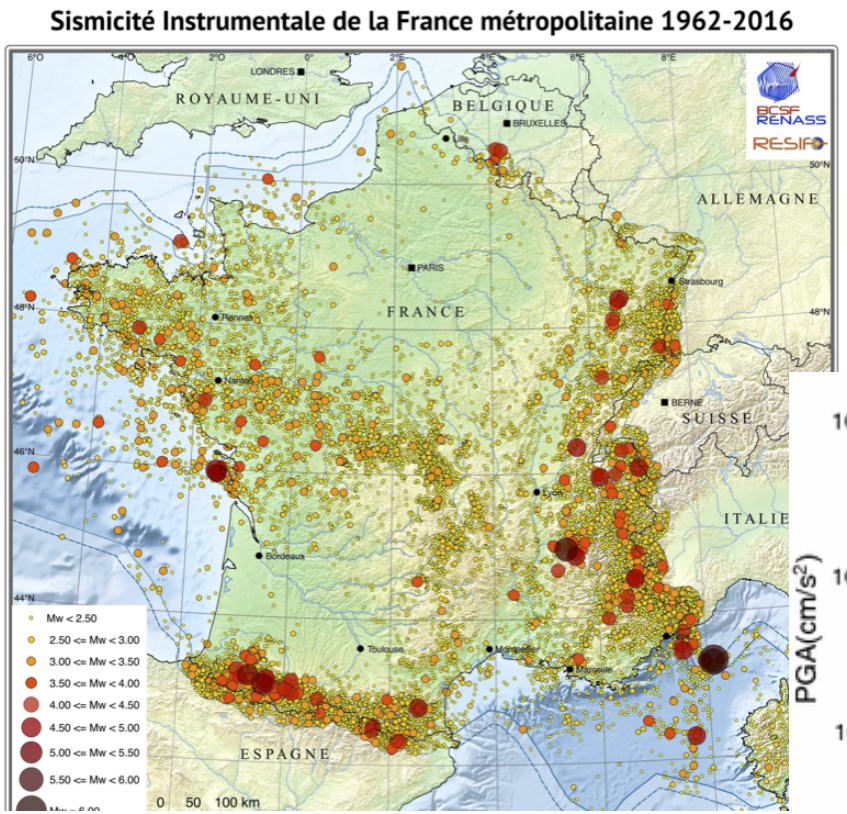
Outline

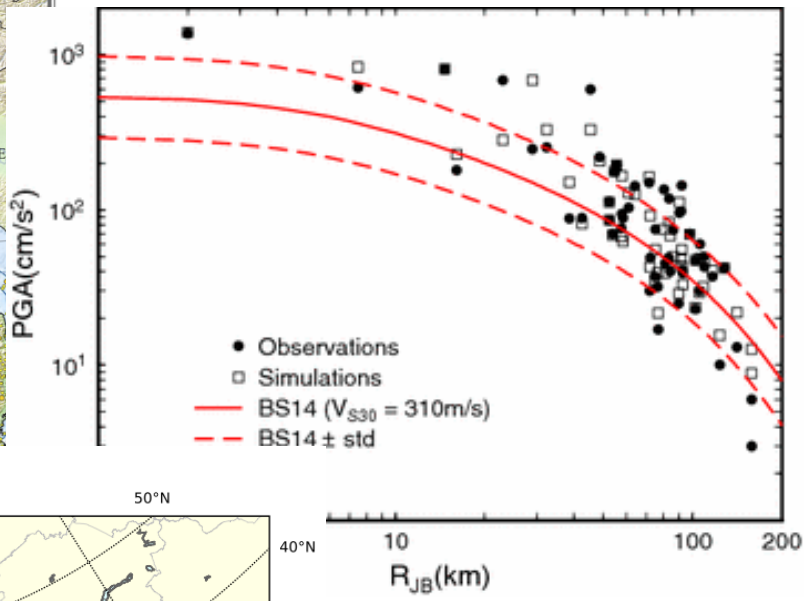
1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

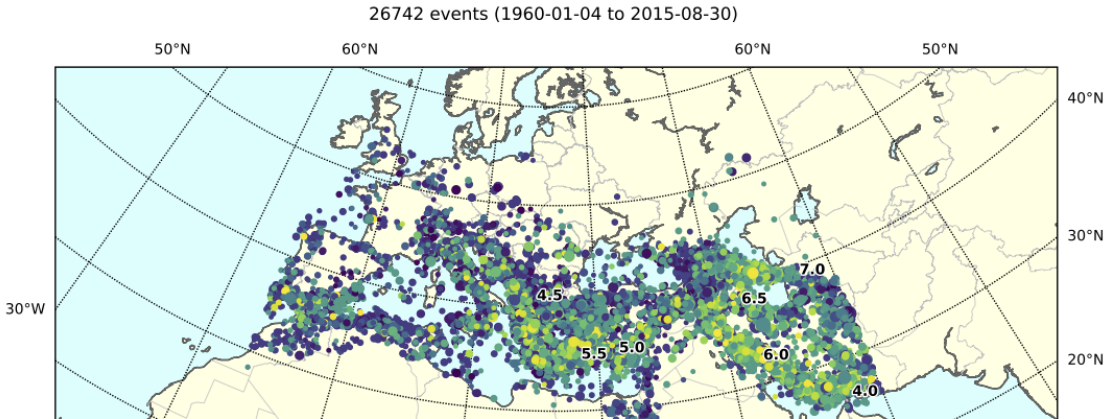
2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Seismic ground motion
Databases

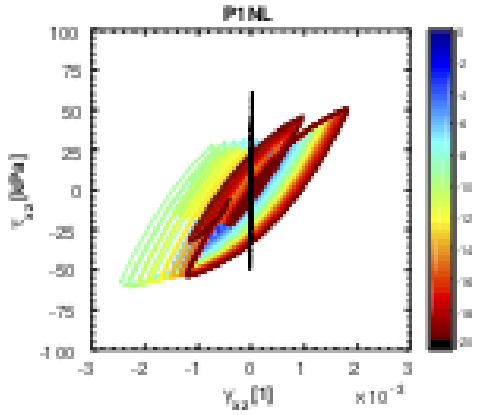


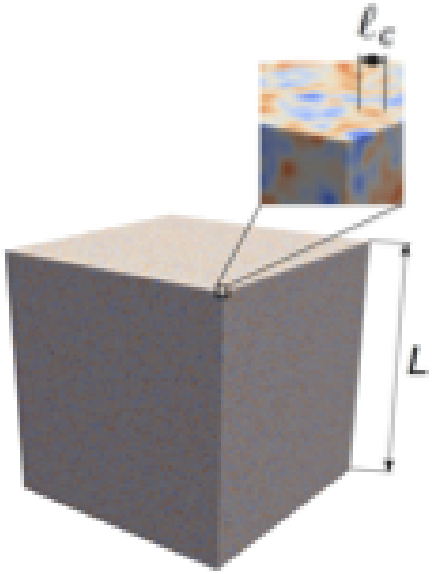


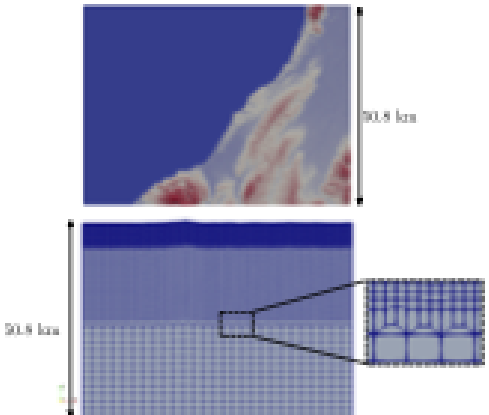


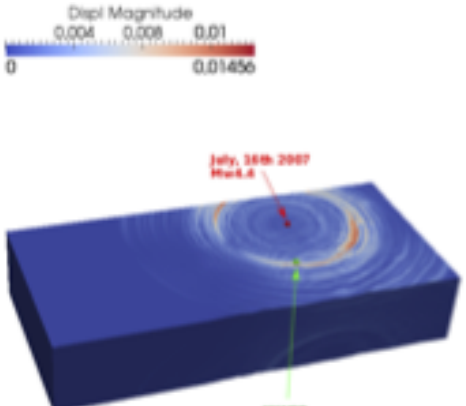
Seismic ground motion
Simulation at Low frequency

- **SEM3D**: Spectral Finite Element, Domain Decomposition, Random fields. Non-linear behavior, extended source model

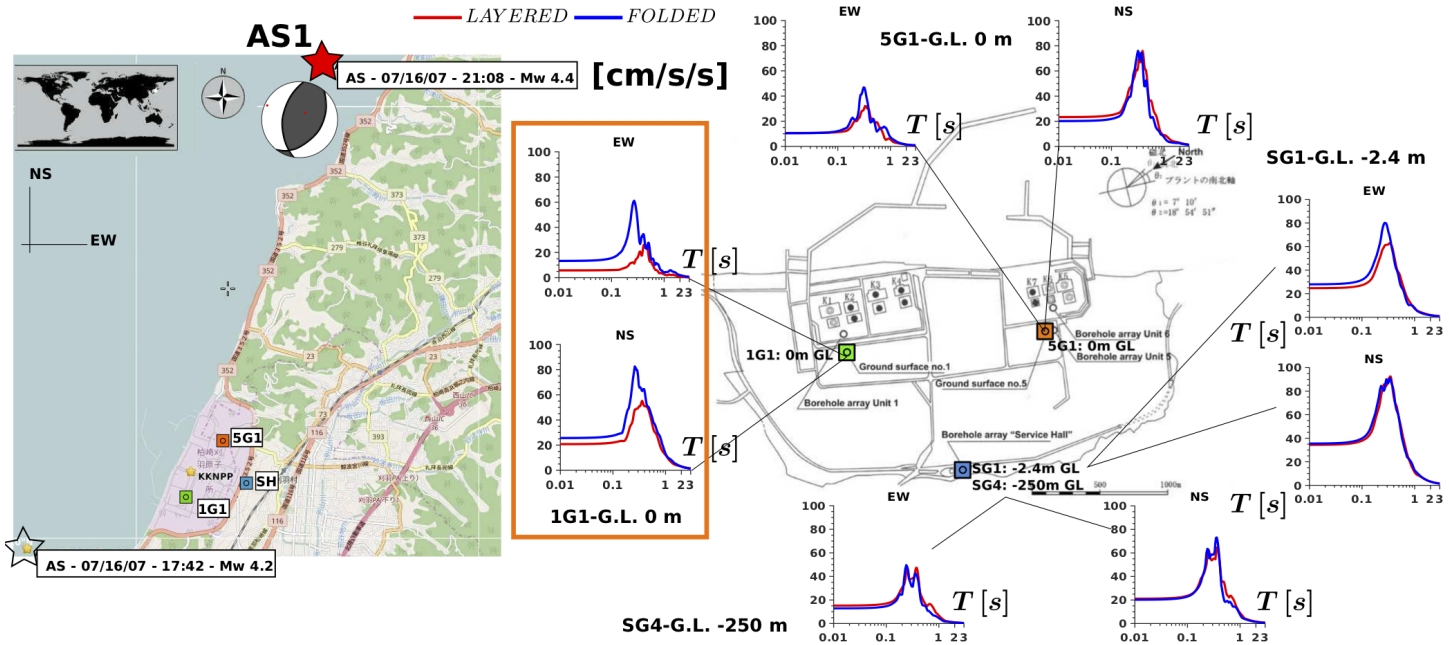








Seismic ground motion
Validation



Outline

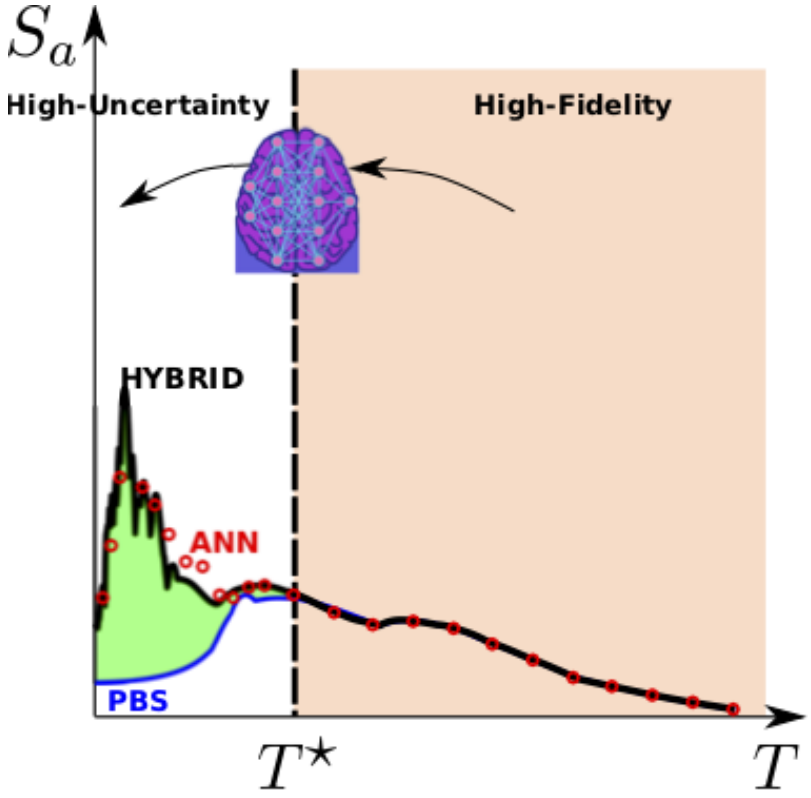
1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Seismic ground motion
Simulation at High Frequency

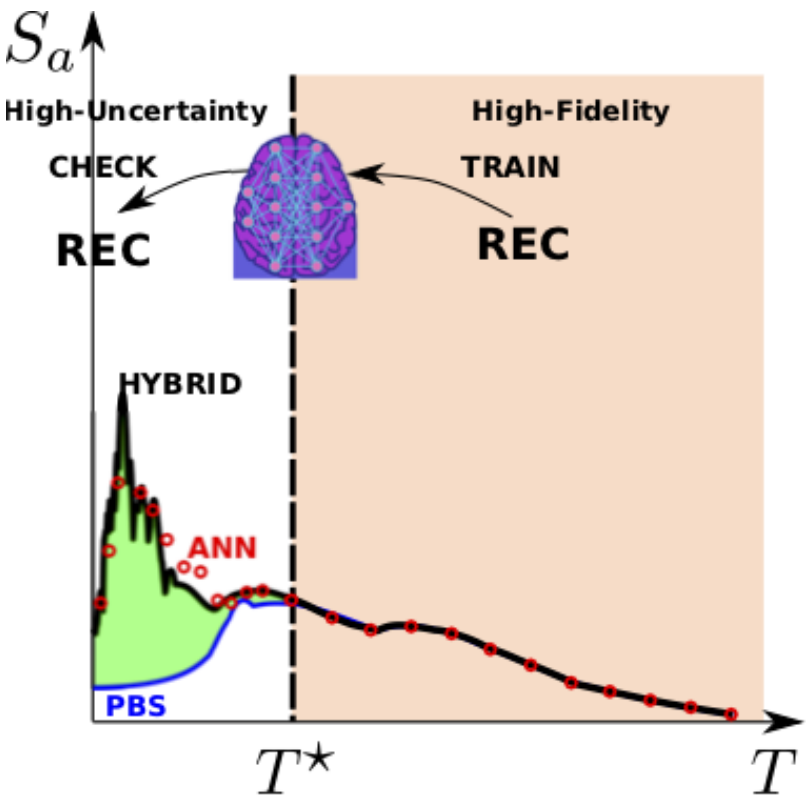
- Mapping short period spectral amplitude Y on long period ones X: $Y = H(X; W)$



$$H = h_1 \circ h_2 \qquad h_k(X) = \sigma_k(W_k X + b_k)$$

Seismic ground motion
Simulation at High Frequency

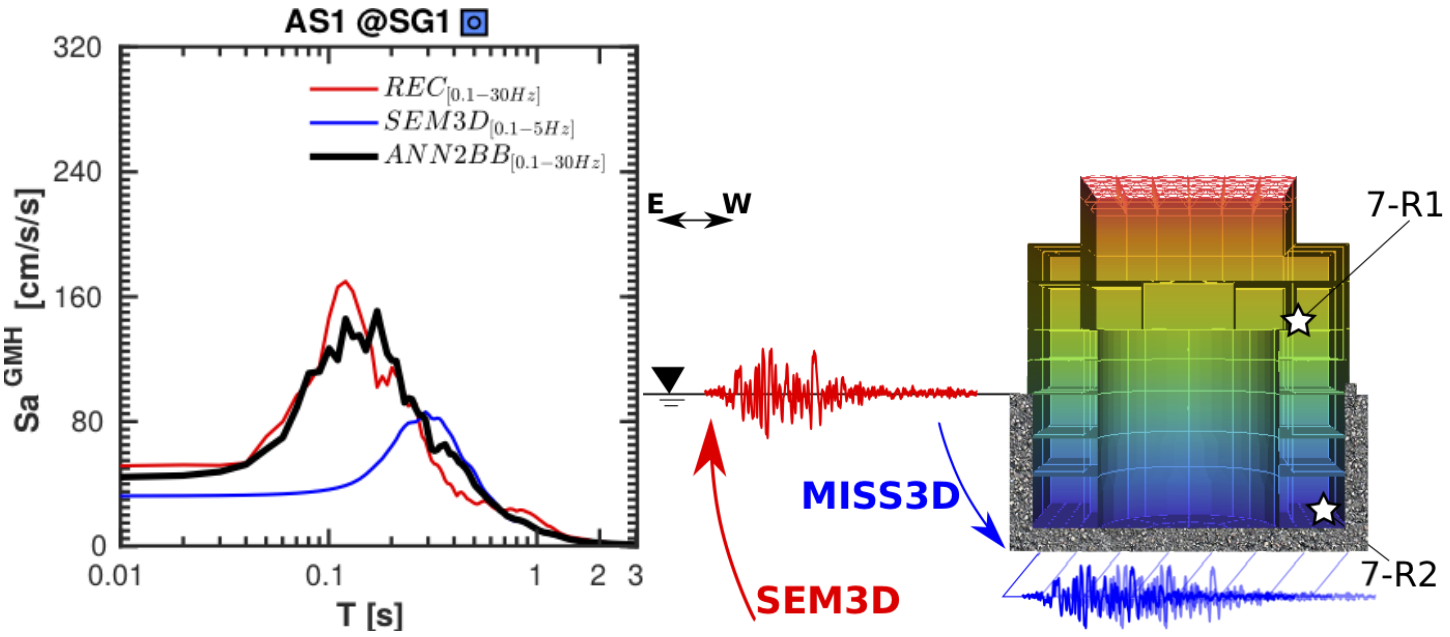
- Training on Database $(X_i, Y_i)_{i=1,m}$ Paolucci, Gatti (BSSA) 2018



$$W^* = \underset{W}{\operatorname{argmin}} \frac{1}{m} \sum_i \|Y_i - H(X_i; W)\|^2$$

Validation

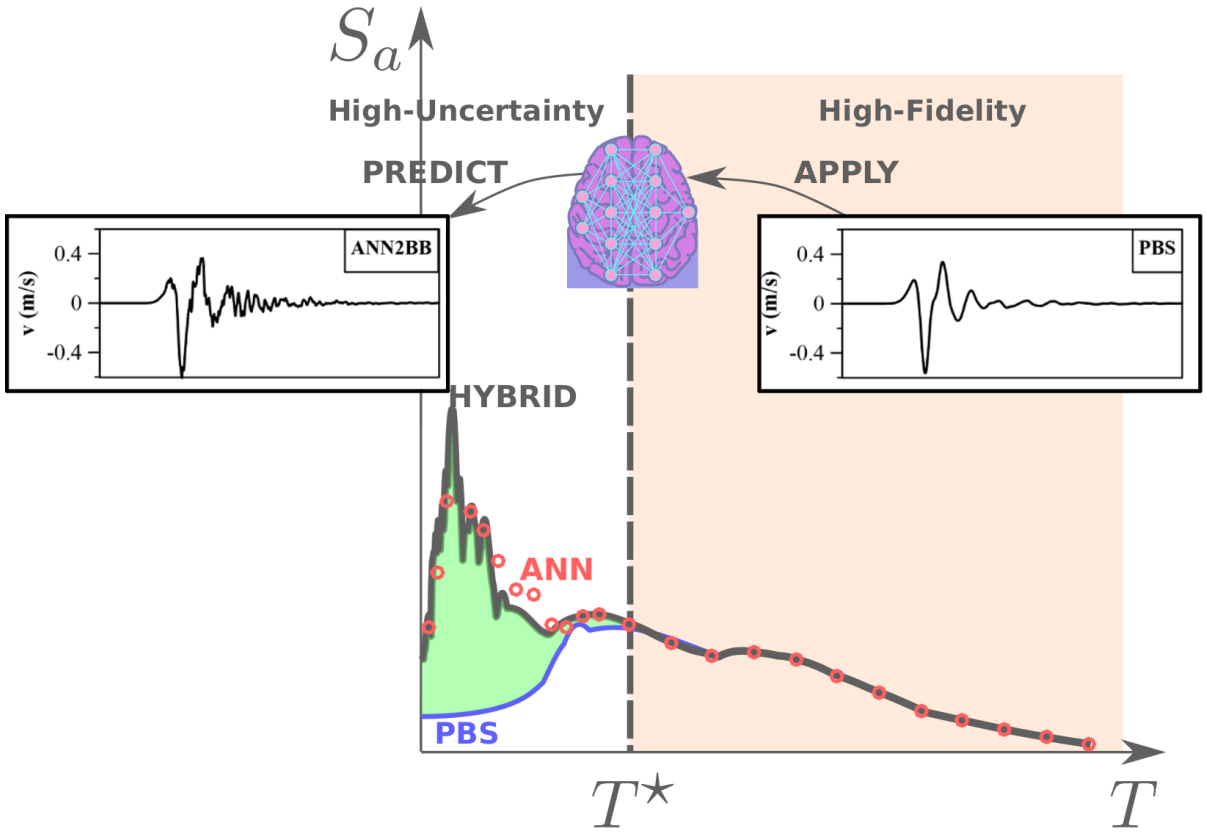
- **Kashiwazaki-Kariwa Nuclear Power Plant** (SINAPS project)

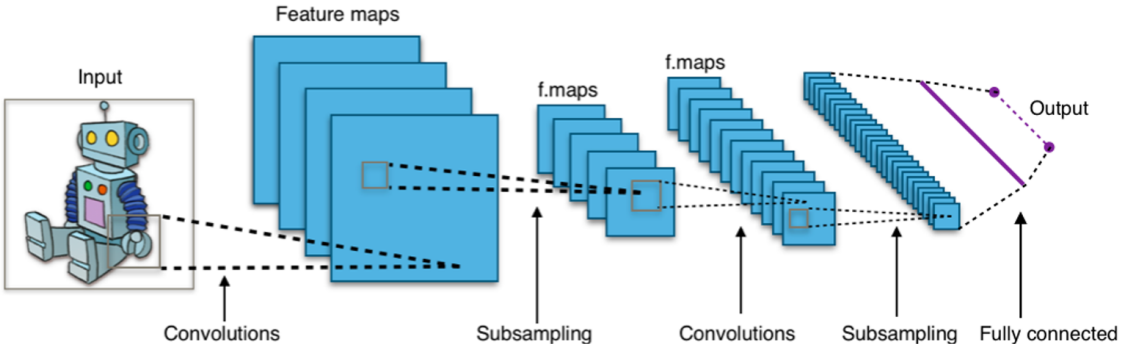


Gatti et Al. (GJI) 2018

Ground Motion simulation

Hybrid Physics-based/Databases signal processing





Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Deterministic view of machine learning

- **Objective**: Identify a function based on samples:

$$\text{Find } \mathbf{h} \in \mathcal{H} \quad | \quad \mathbf{y} \approx \mathbf{h}(\mathbf{x}) \qquad \text{knowing } \mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,m}$$

  - **Input**: $\mathbf{x} \in \mathcal{X}$ a vector of size d (real or integer)
  - **Ouput**: $\mathbf{y} \in \mathcal{Y}$ a vector of:
    - **Labels** ($y_k \in \{"A", "B", ...\}$) : *Classification*
    - **Numbers** ($y_k \in \mathbb{R}$ or $\mathbb{Z}$): *Regression*
  - **Hypothesis**: $\mathbf{h}$ a hypothetical function in class $\mathcal{H}$

$$\mathbf{h}(\mathbf{x}) = \sigma\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right)$$

  - **Sample set**: $\mathcal{S}$ of size m
  - **Deep Networks**: $\mathbf{h} = \mathbf{h}_n \circ ... \circ \mathbf{h}_1$
  - **Convolution Networks**: $W_{p+s,q+s} = W_{p,q}$
  - *Universal Approximation Theorem*

Empirical risk minimisation

- **Error/Risk/Loss function**

$$l(\mathbf{y}, \mathbf{h}(\mathbf{x})) \in \mathbb{R}^+ \qquad \text{with} \qquad l(\mathbf{y}, \mathbf{y}) = 0$$

  - **Classification** $l = \sum_k l_k$ with:
    - $l_k = 1$ when $y_k \neq h_k(\mathbf{x})$,
    - $l_k = 0$ when $y_k = h_k(\mathbf{x})$
  - **Regression**

$$l = \|\mathbf{y} - \mathbf{h}(\mathbf{x})\|_p^p$$

- **Empirical Risk**

$$L_S(\mathbf{h}) = \frac{1}{m} \sum_i l(\mathbf{y}_i, \mathbf{h}(\mathbf{x}_i))$$

- **Empirical Risk minimisation**

$$\hat{\mathbf{h}}_S = \underset{\mathbf{h} \in \mathcal{H}}{\text{argmin}}\ L_S(\mathbf{h})$$

Probabilistic approach

- Unknown probability $P(\mathrm{X},\mathrm{Y})$ on $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$

$$P_{\mathrm{X}}(\mathrm{X}) = \int_{\mathcal{Y}} dP(\mathrm{X},\mathrm{Y}), \qquad P_{\mathrm{Y}|\mathrm{X}}(\mathrm{Y}|\mathrm{X}) = \frac{P(\mathrm{X},\mathrm{Y})}{P_{\mathrm{X}}(\mathrm{X})}$$

  - **Problem**: Find the *conditional probability* knowing $\mathcal{S}$

$$\hat{Q}(\mathrm{Y}|\mathrm{X}) \approx P_{\mathrm{Y}|\mathrm{X},\mathcal{S}}(\mathrm{Y}|\mathrm{X},\mathcal{S})$$

  - $\mathcal{S}$ being identically sampled: $P_{So}(\mathrm{X},\mathrm{Y}) = \frac{\mathbb{1}_{\mathcal{S}}(\mathrm{X},\mathrm{Y})}{m}$

$$Q_{So}(\mathrm{Y}|\mathrm{X}) = \frac{\mathbb{1}_{\mathcal{S}}}{m}\frac{1}{P_{So}(\mathrm{X})}, \qquad P_{So}(\mathbf{x}) = \frac{\sum_i \mathbb{1}_{\{\mathbf{x}=\mathbf{x}_i\}}}{m}$$

- **Empirical error**

$$L_{\mathcal{S}}(\hat{Q}) = \overbrace{\int_{\mathcal{D}} \ln\left(\frac{Q_{So}}{\hat{Q}}\right) P_{So} d\mathrm{X} d\mathrm{Y}}^{D_{\mathrm{KL}}(Q_{So}\|\hat{Q})} = \overbrace{\frac{-1}{m}\sum_i \ln \hat{Q}(\mathrm{Y}_i|\mathrm{X}_i)}^{\to E[-\ln \hat{Q}]}$$

Probabilistic approach
Link with the deterministic case

- **From empirical error to conditional probability**

$$Q(\mathbf{y}|\mathbf{x}) = \frac{e^{-l(\mathbf{y},\mathbf{h}(\mathbf{x}))}}{Z(\mathbf{x})}, \qquad Z(\mathbf{x}) = \int_{\mathcal{Y}} e^{-l(\mathbf{y},\mathbf{h}(\mathbf{x}))} d\mathbf{y}$$

  - Classification: $l(\mathbf{y}, \mathbf{h}(\mathbf{x})) = \mathbb{1}_{\mathbf{y} \neq \mathbf{h}(\mathbf{x})}$

$$Q(\text{"}A\text{"}|\mathbf{x}) = \frac{e^{-\mathbb{1}_{\{\text{"}A\text{"} \neq \mathbf{h}(\mathbf{x})\}}}}{Z(\mathbf{x})}, \qquad Z(\mathbf{x}) = \sum_{\mathbf{y}' \in \{\text{"}A\text{"},\ldots\}} e^{-\mathbb{1}_{\{\mathbf{y}' \neq \mathbf{h}(\mathbf{x})\}}}$$

  - L_2 Regression

$$L_D(\mathbf{h}) = E\left[L_S(Q)\right] = E\left[\frac{\|\mathbf{y} - \mathbf{h}(\mathbf{x})\|^2}{2\sigma^2}\right] + S_X(\mathbf{h})$$

Deep neural network
From data to probability

- **Converting scores to probability**
  - **Softmax** for multiple features output with individual scores $s_j(\mathbf{x})$ and total score $s = \sum_j s_j$

$$Q(y=k|\mathbf{x}) = \frac{e^{-(s-s_k)}}{\sum_j e^{-(s-s_j)}} = \frac{1}{\sum_j e^{s_j - s_k}}$$

  - **Discriminator** D for an unique feature with score $s(\mathbf{x})$ of not having the feature:

$$D(\mathbf{x}) = Q(y = \text{"0"}|\mathbf{x}) = \frac{1}{1+e^{-s}}$$

$$\begin{aligned} L_S(D) &\overset{m\to+\infty}{\to} E_{y\mathbf{x}}[-\ln Q(y|\mathbf{x})] = S(y|X) \\ &= E_{\mathbf{x}|y=\text{"0"}}[-\ln D] + E_{\mathbf{x}|y\neq\text{"0"}}[-\ln(1-D)] \\ \hat{D}(\mathbf{x}) &= \underset{D}{\operatorname{argmin}}\ L_S(D) \end{aligned}$$

Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Auto-encoders
And convolution networks

- **Autoencoder**

$$\mathbf{z} = F(\mathbf{x}), \qquad \mathbf{y} = G(\mathbf{z}), \qquad S = \{\mathbf{x}_i, \mathbf{y}_i = \mathbf{x}_i\}_{i=1,m}$$

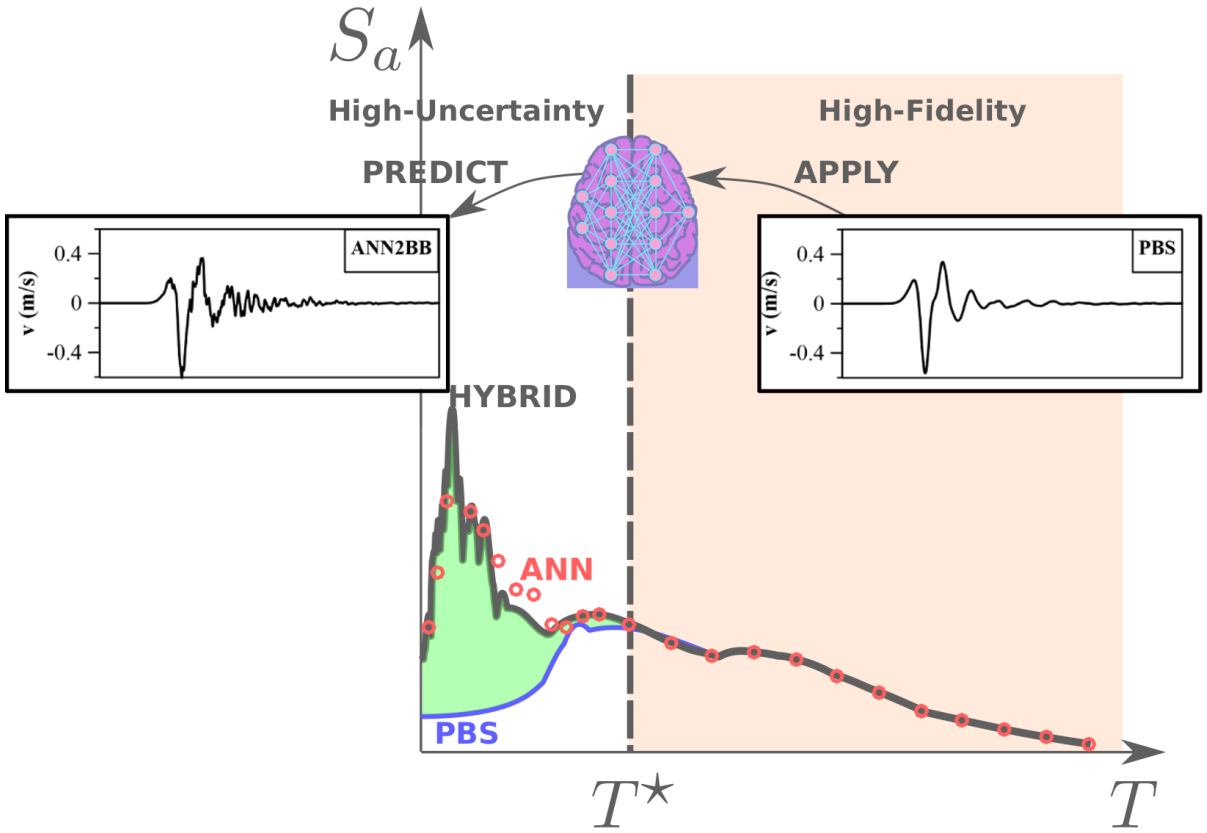
$$L_S(F,G) = \sum_i \|\mathbf{x}_i - F(G(\mathbf{x}_i))\|^2$$

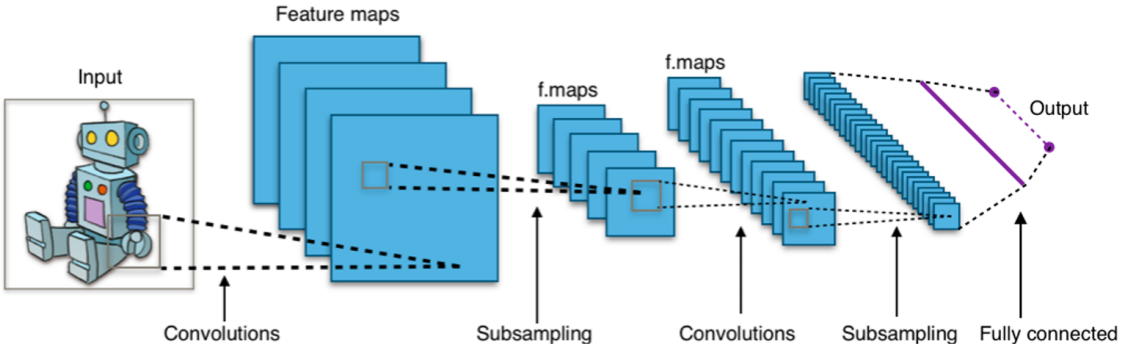
$$(\hat{F}, \hat{G}) = \underset{F,G}{\text{argmin}}\ L_S(F,G)$$

  - $\mathcal{Z} = F(\mathcal{X})$ is smaller than $\mathcal{X}$
  - Noise is added to the input
  - Weights are linked when $F = G^{-1}$: $\mathbf{W}_F = \mathbf{W}_G^T$
  - F and G can have several layers and be convolutional
- *Remark:* In the linear case
  - **Convolutional case**: F is any Wavelet transform
  - **full case**: G is any Karhunen-Loeve or POD
  - *Weakness*: P_{Z} is not simple

Ground Motion simulation

Hybrid Physics-based/Databases signal processing





Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Generative Adversarial Networks (GAN)
Conditional Entropy

- **Mixture Probability of**
  - $\mathbf{x}_1 \in \mathcal{X}$ with distribution p_1
  - $\mathbf{x}_2 \in \mathcal{X}$ with distribution p_2.
  - $\mathbf{y} \in \{"1", "2"\}$ with respective probability α and $1-\alpha$
  - $(\mathbf{x}, \mathbf{y})$ with $p(.|\mathbf{y} = "1") = p_1$ and $p(.|\mathbf{y} = "2") = p_2$

$$p(\mathbf{x}) = \alpha p_1(\mathbf{x}) + (1-\alpha) p_2(\mathbf{x})$$

  - $\hat{D}(\mathbf{x}) = p(\mathbf{y} = "1"|\mathbf{x})$
- Entropy

$$S(\mathbf{x}, \mathbf{y}) = \underbrace{S(\mathbf{x}|\mathbf{y})}_{\alpha S(\mathbf{x}_1) + (1-\alpha) S(\mathbf{x}_2)} + S(\mathbf{y}) = S(\mathbf{y}|\mathbf{x}) + S(\mathbf{x})$$

- Cross-Information

$$\begin{aligned} I(\mathbf{x}, \mathbf{y}) &= S(\mathbf{y}) - S(\mathbf{y}|\mathbf{x}) \geq 0 \\ S(\mathbf{y}|\mathbf{x}) &= \alpha E_{x_1}\left(-\ln \hat{D}\right) + (1-\alpha) E_{x_2}\left(-\ln(1-\hat{D})\right) \end{aligned}$$

Generative Adversarial Networks (GAN)

Jensen-Shannon Divergence

- **Mixture Probability of**
  - $\mathbf{x}_1 \in \mathcal{X}$ with distribution p_1
  - $\mathbf{x}_2 \in \mathcal{X}$ with distribution p_2.
  - $\mathbf{y} \in \{1, 2\}$ with respective probability α and $1 - \alpha$
  - $(\mathbf{x}, \mathbf{y})$ with $p(\mathbf{x}|\mathbf{y} = 1) = p_1(\mathbf{x})$ and $p(\mathbf{x}|\mathbf{y} = 2) = p_2(\mathbf{x})$

$$p(\mathbf{x}) = \alpha p_1(\mathbf{x}) + (1 - \alpha) p_2(\mathbf{x})$$

  - $D(\mathbf{x}) = p(\mathbf{y} = 1|\mathbf{x})$
- **Jensen-Shannon Divergence** as cross information:

$$\begin{aligned} S(\mathbf{y}) - S(\mathbf{y}|\mathbf{x}) &= S(\mathbf{x}) - S(\mathbf{x}|\mathbf{y}) = \\ D_{\mathrm{JS}}(\mathbf{x}_1 \| \mathbf{x}_2) &= \alpha E_{\mathbf{x}_1}\left[\ln \frac{p_1}{p}\right] + (1 - \alpha) E_{\mathbf{x}_2}\left[\ln \frac{p_2}{p}\right] \\ &= \alpha D_{\mathrm{KL}}(p_1 \| p) + (1 - \alpha) D_{\mathrm{KL}}(p_2 \| p) \end{aligned}$$

Generative Adversarial Networks (GAN)

Empirical error I

- **Finding the best model $\mathbf{x}_m$ for data $\mathbf{x}_d$**

$$\hat{\mathbf{x}}_m = \underset{\mathbf{x}_m}{\operatorname{argmin}}\ D_{\mathrm{JS}}(\mathbf{x}_d \| \mathbf{x}_m)$$

  - Finding $D_{\mathrm{JS}}(\mathbf{x}_d \| \mathbf{x}_m)$ estimate based on empirical data
  - Finding a minimum (Stochastic gradient).

- $(-D_{\mathrm{JS}})$ as the entropy of conditional probability [1] $p(\mathbf{y} \in \{"d", "m"\} | \mathbf{x})$:

$$S(\mathbf{y}|\mathbf{x}) \quad = \quad -D_{\mathrm{JS}}(\mathbf{x}_d \| \mathbf{x}_m) + S(Y)$$

[1]

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} = \alpha \frac{p(\mathbf{x}_d)}{p(\mathbf{x})} \text{ or } (1-\alpha)\frac{p(\mathbf{x}_m)}{p(\mathbf{x})}$$

Generative Adversarial Networks (GAN)

Empirical error II

- D_{JS} as the conditional probability $p(\mathbf{y}\in\{"d","m"\}|\mathbf{x})$

$$S(\mathbf{y}|\mathbf{x}) \quad = \quad -D_{\mathrm{JS}}(\mathbf{x}_d\|\mathbf{x}_m) + S_\alpha$$

- Find a discriminator $\hat{D}(\mathbf{x})$ such that $\hat{D}(\mathbf{x}) = P("d"|\mathbf{x})$:

$$S(\mathbf{y}|\mathbf{x}) = -\alpha E\left[\ln \hat{D}\right] - (1-\alpha)E\left[\ln(1-\hat{D})\right]$$

- Estimation of D as a **discriminator** with a classification Neural Network (max Log Likelihood):

$$\hat{D} = \underset{D}{\mathrm{argmax}}\ \frac{1}{m}\sum_i \ln D(\mathbf{x}_i)$$

Generative Adversarial Networks (GAN)
Implementation

- **A random unit-centered-Gaussian vector z**
- **The generator**: $\mathbf{x}_m = G(\mathbf{z})$
- **The Discriminator**: $D(\mathbf{x}) \in [0,1]$
- **The mixture sampler**: $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1,m}$ with $\mathbf{x}_i \in \{\mathbf{x}_d, G(\mathbf{z})\}$ with probability α and $1-\alpha$

$$\hat{G} = \underset{G}{\operatorname{argmin}} \max_{D} \quad \alpha E_d \left[\ln D(\mathbf{x}_d)\right] + (1-\alpha) E_z \left[\ln(1 - D(G(\mathbf{z}))]\right)$$

- **Properties**
  - $\hat{G}$ minimise the distance (JS-Divergence) between the modeled and the data distributions
  - $\hat{G}$ maximise the conditional entropy $S(Y|X)$

Goodfellow 2014

Generative Adversarial Networks (GAN)

Strength and weakness

$$\hat{G} = \underset{G}{\operatorname{argmin}} \max_{D} \quad \alpha E_d \left[\ln D(\mathbf{x}_d)\right] + (1-\alpha) E_m \left[\ln(1 - D(G(\mathbf{z}_m)))\right])$$

- **Strength**
  - Identification of a representation model of sampled data using independent unit random seeds $\mathbf{z}$
  - A Polynomial Chaos Expansion (PCE) can be built when $\mathbf{z}$ is a unit Gaussian random vector

$$\mathbf{x}_m = \sum_j \mathbf{x}_j \psi_j(\mathbf{z}) \quad , \quad \mathbf{x}_j = \int G(\mathbf{z}) \psi_j(\mathbf{z}) \nu(\mathbf{z}) d\mathbf{z}$$

- **Weaknesses** (same as PCE)
  - Non uniqueness
  - $\mathbf{z}$ is a collection of hidden features
  - No encoding algorithm ($\mathbf{z} = F(\mathbf{x})$)

Adversarial Encoder (GAN)

- Mixture between $\mathbf{z}_d = F(\mathbf{x}_d)$ and unitary $\mathbf{z}_m$ with $D_z = P(Y = "m"|\mathbf{z})$

$$\begin{aligned} -S(Y|Z) &= (1-\alpha)E_m(\ln D_z) + \alpha E_d(\ln(1 - D_z(F(.))) \\ &= D_{JS}(p_{z_m}||p_{z_d}) - S(Y) \end{aligned}$$

$$\begin{aligned} \hat{F} = \underset{F}{\operatorname{argmin}} \max_{D_z} \quad & \alpha E_d\left[\ln(1 - D_z(F(\mathbf{x}_d)))\right] \\ + \quad & (1-\alpha)E_z\left[\ln D_z\right] \end{aligned}$$

- **Weakness**
  - Marginals on $\mathbf{x}$ and $\mathbf{z}$ are equally distributed but their joint probabilities are not. Hence:

$$\mathbf{x}_d \neq G(F(\mathbf{x}_d)) \quad , \quad \mathbf{z} \neq F(G(\mathbf{z}))$$

  - No link between encoder and generator when learning.

Bidirectional GAN

- **biGAN**: $\hat{D}_{xz}(\mathbf{x},\mathbf{z}) \approx P(\text{"}m\text{"}|(\mathbf{x},\mathbf{z}))$

$$
\begin{aligned}
(\hat{F},\hat{G},\hat{D}_{xz}) &= \underset{F,G}{\operatorname{argmax}}\ \underset{D_{xz}}{\operatorname{argmin}}\ L_S(D_{xz},F,G) \\
L_S(D_{xz},F,G) &= \frac{-1}{m}\sum_i (\ln D_{xz}(G(\mathbf{z}_i),\mathbf{z}_i))\mathbb{1}_{y_i=\text{"}m\text{"}} \\
&\quad + \ln(1-D_{xz}(\mathbf{x}_i,F(\mathbf{x}_i)))\mathbb{1}_{y_i=\text{"}d\text{"}}) \\
&\approx S(Y|X,Z)
\end{aligned}
$$

- **Weakness**
  - Sample set does not cover all $\mathcal{X}\times\mathcal{Z}$ space

$$\mathbf{x}_d \neq G(F(\mathbf{x}_d)) \qquad , \qquad \mathbf{z} \neq F(G(\mathbf{z}))$$

  - Need to better control $S(X|Z)\to 0$ or $\mathbf{x}_d - G(F(\mathbf{x}_d)$.

Donahue, Dumoulin 2017

Cycling GAN
ALICE

- $\hat{D}_{xz}(\mathbf{x}, \mathbf{z}) \approx P(\text{"}m\text{"}|(\mathbf{x}, \mathbf{z}))$

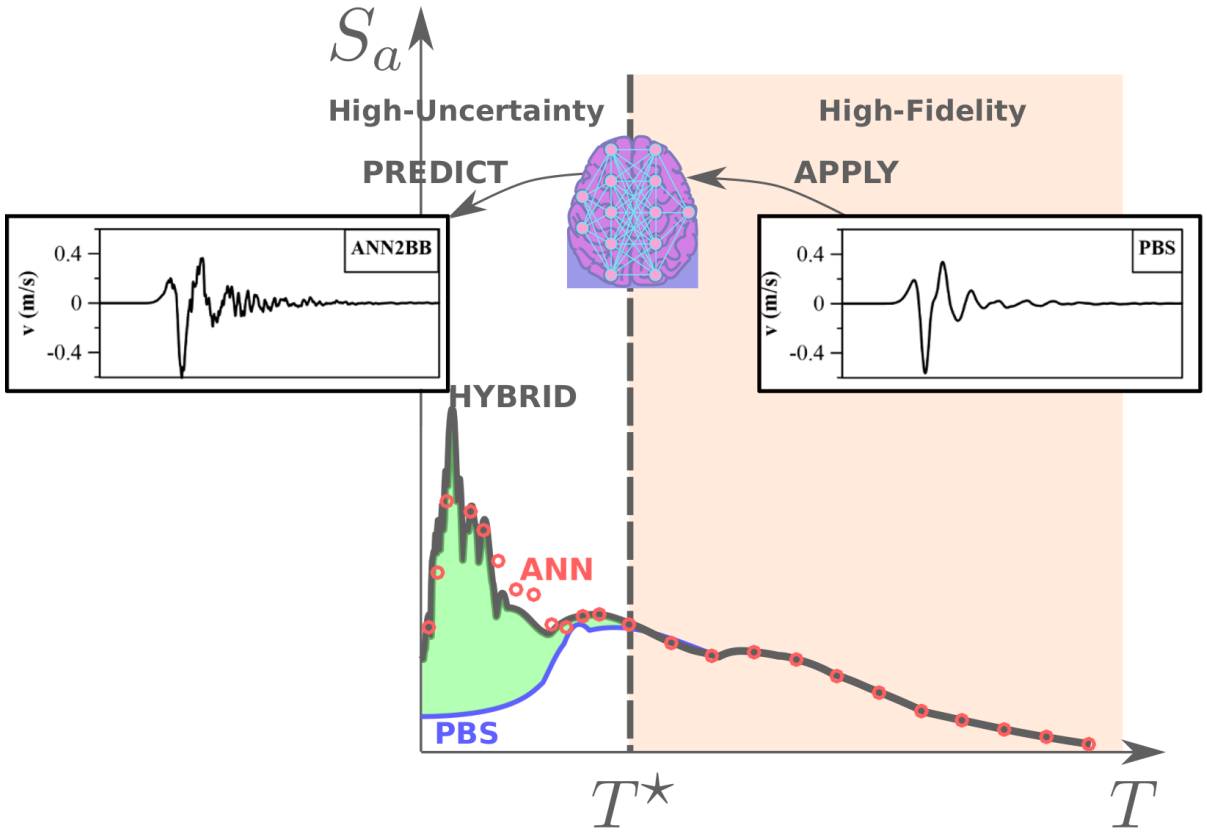
$$
\begin{aligned}
(\hat{F}, \hat{G}, \hat{D}_{xz}) &= \underset{F,G}{\operatorname{argmax}} \ \underset{D_{xz}}{\operatorname{argmin}} \ L_S^*(D_{xz}, F, G) \\
L_S^*(D_{xz}, F, G) &= L_S(D_{xz}, F, G) + \eta \|\mathbf{x}_d - G(F(\mathbf{x}_d))\|_p^p
\end{aligned}
$$

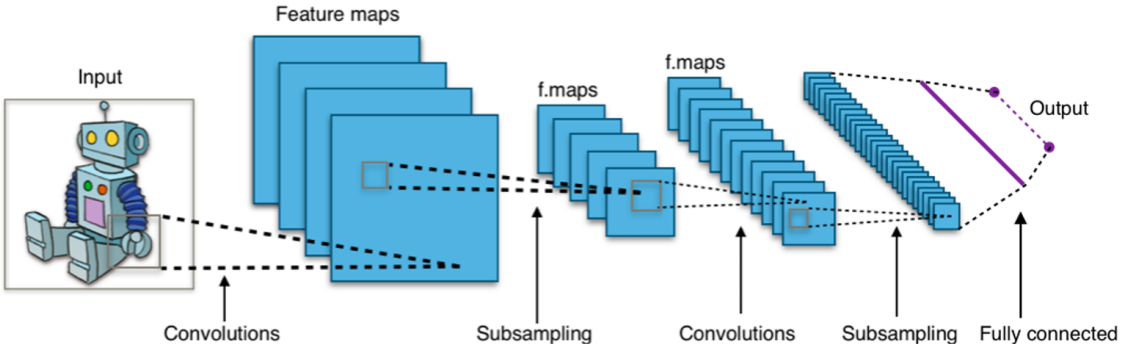
- **Strength**
  - It works (!)
  - Filtering $\|\mathbf{x}_d - G(F(\mathbf{x}_d)\|_p^p$ gives broadband output for filtered inputs.
  - Use of GAN to discriminate $\mathbf{x}_d \neq G(F(\mathbf{x}_d))$.

Li et Al. 2017

Ground Motion simulation

Hybrid Physics-based/Databases signal processing





Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Adversarial Generative auto-encoders

- **Encoder**: $\mathbf{z}_1 = F(\mathbf{x})$
- **Generator**: $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ with $\mathbf{z}_2$ Gaussian

$$\mathbf{x}_m = G(\mathbf{z})$$

- **Optimisation**

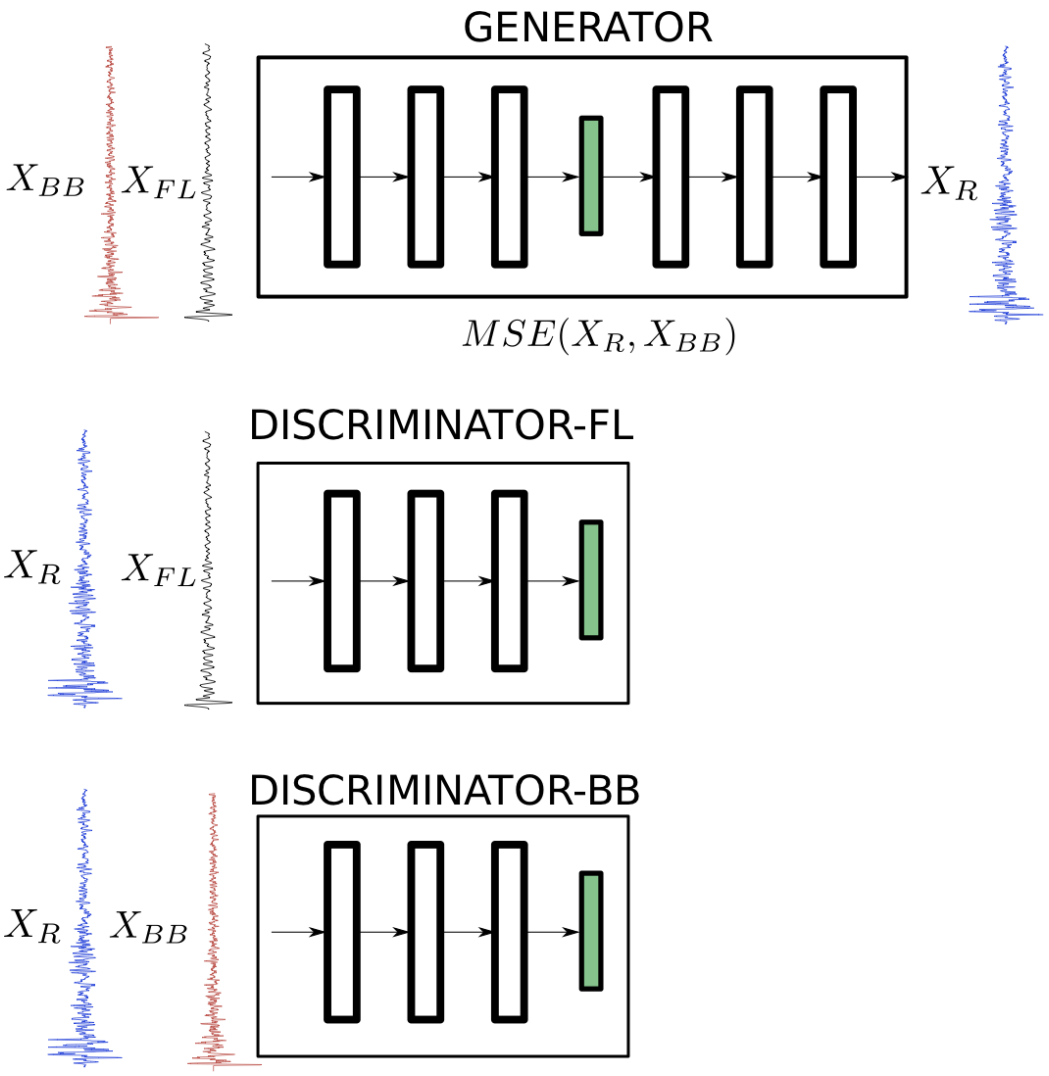
$$(\hat{F}, \hat{G}) = \underset{(G,F)}{\operatorname{argmin}} \underbrace{L_S(G,F)}_{\frac{1}{m}\sum_i \|\mathbf{x}_i - G(\mathbf{z}_i)\|^2}$$

  - D discriminator between data $\mathbf{x}_d$ and model $\mathbf{x}_m$:

$$\begin{aligned}(\hat{F}, \hat{G}) &= \underset{(G,F)}{\operatorname{argmin}} \left(L_S(G,F)\right. \\ &\quad + \max_D \left(E_d\left[\ln D(\mathbf{x}_d)\right] + E_m\left[\ln D(G(\mathbf{z}))\right]\right)\end{aligned}$$

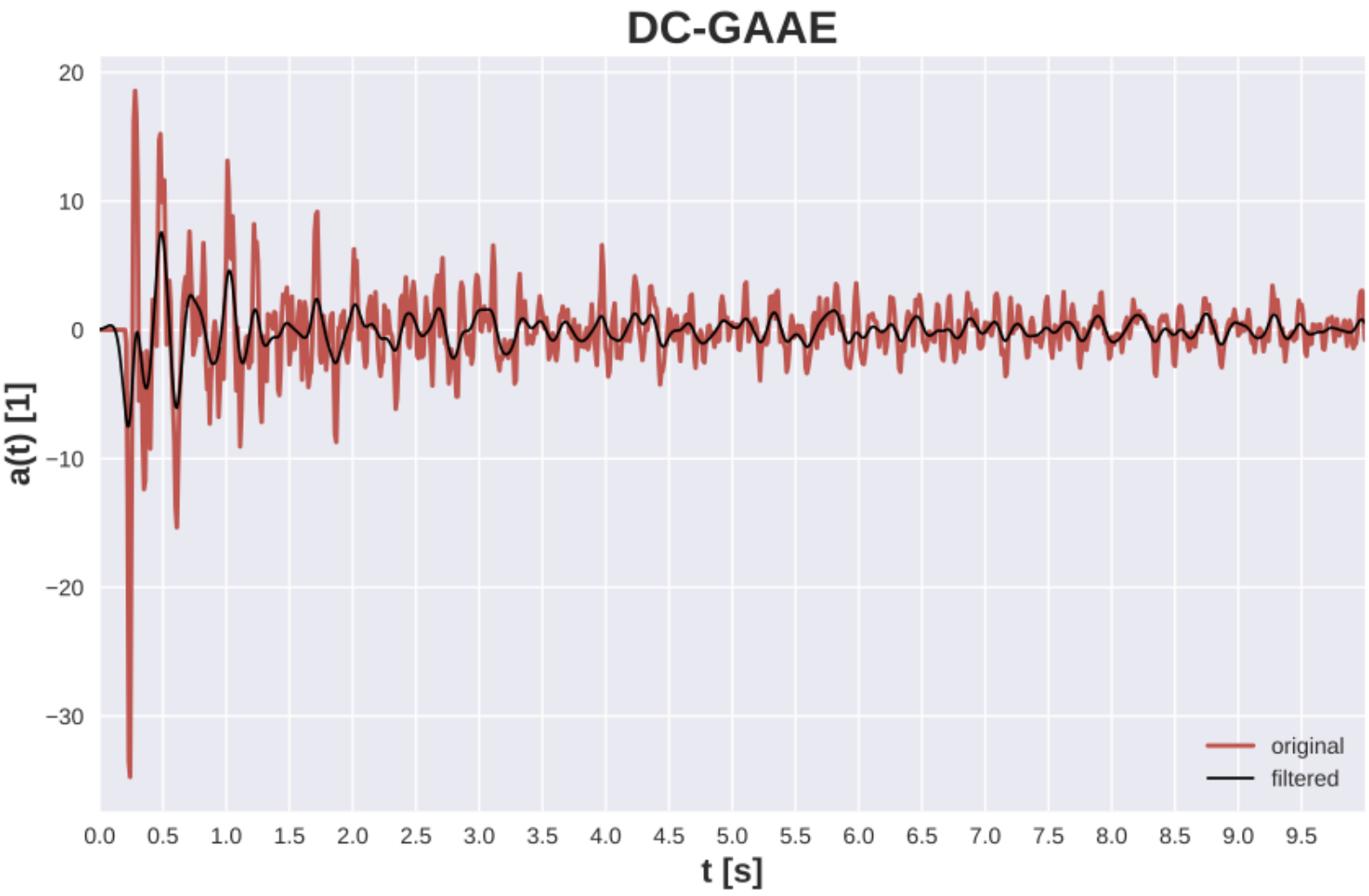
Ground Motion simulation

Hybrid Physics-based/Databases Adversarial Generative auto-encoders



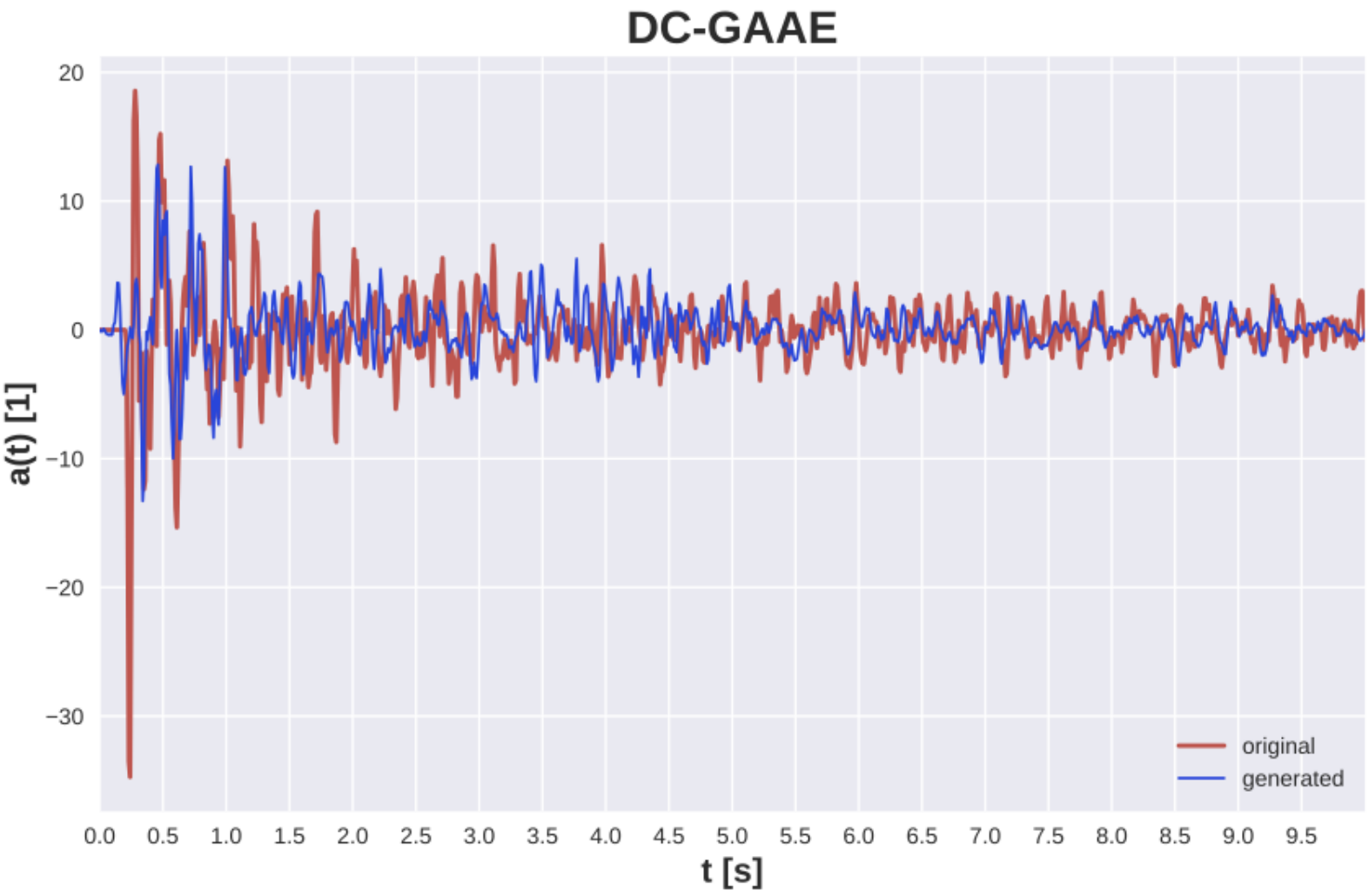
Ground Motion simulation

Hybrid Physics-based/Databases Adversarial Generative auto-encoders



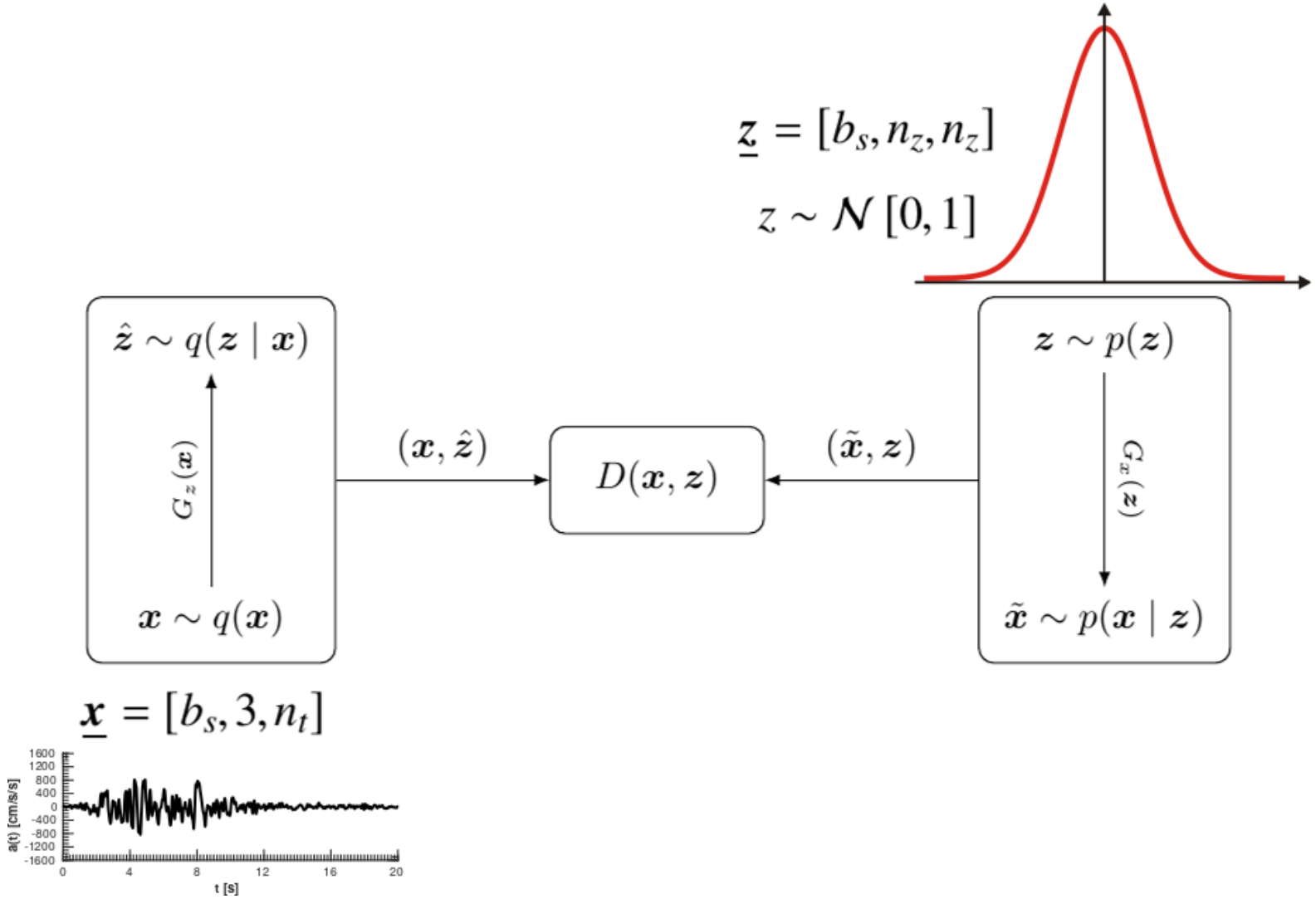
Ground Motion simulation

Hybrid Physics-based/Databases Adversarial Generative auto-encoders



Ground Motion simulation

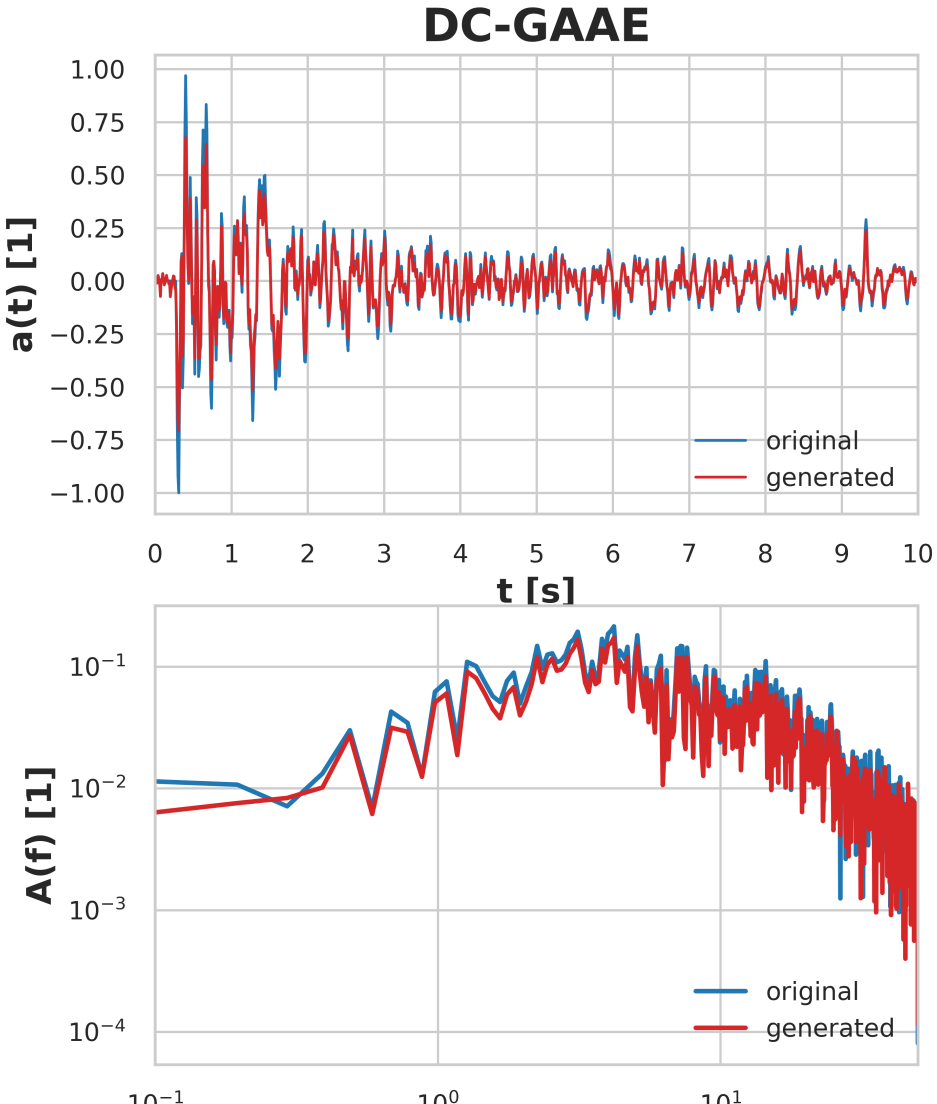
Hybrid Physics-based/Databases signal processing: ALICE like



Ground Motion simulation

Hybrid Physics-based/Databases signal processing: ALICE like

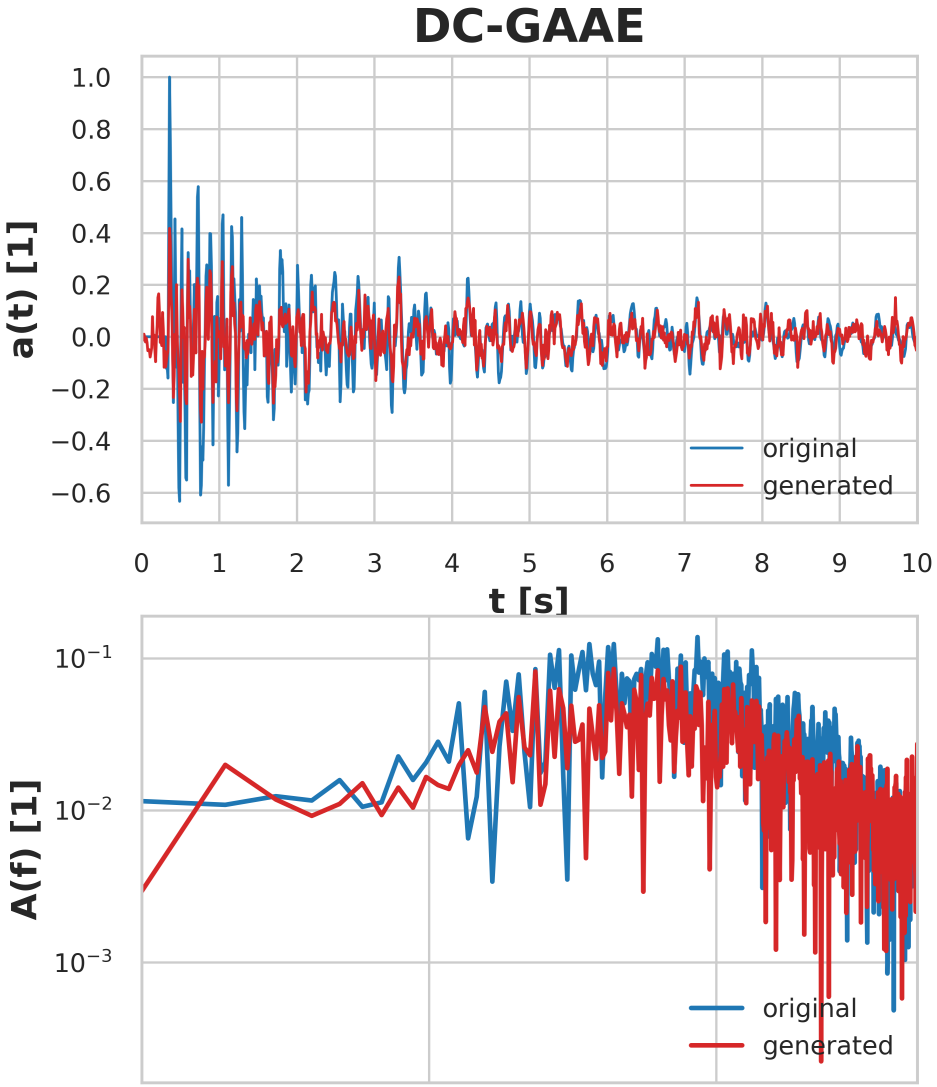
Signal reconstruction



Ground Motion simulation

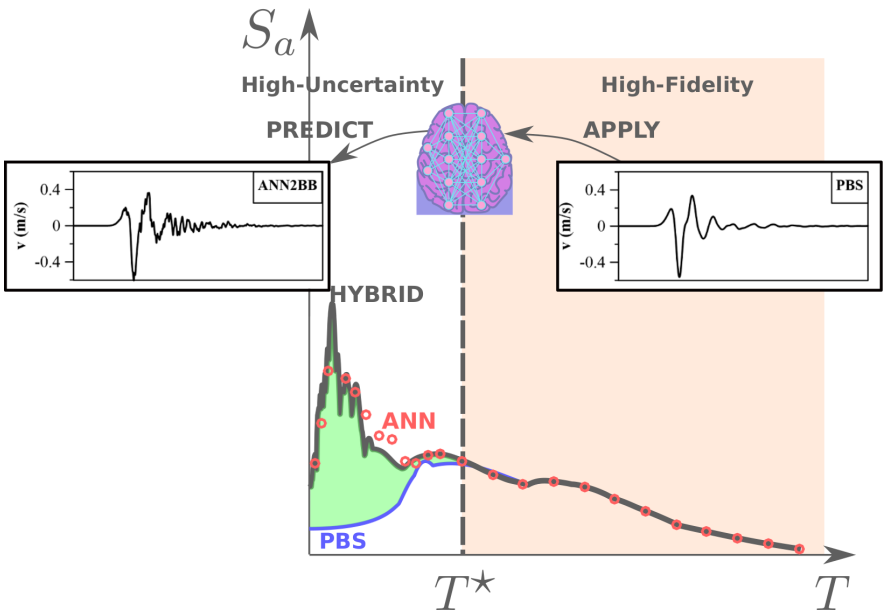
Hybrid Physics-based/Databases signal processing: ALICE+

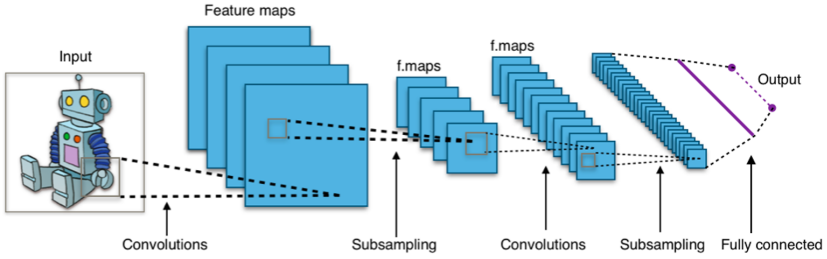
Signal generation from filtered input



Ground Motion simulation

Hybrid Physics-based/Databases signal processing





Thank you !

Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Empirical risk minimisation
Discussion

- **Empirical Risk minimisation**

$$\hat{\mathbf{h}}_S = \underset{\mathbf{h}\in\mathcal{H}}{\operatorname{argmin}}\ L_S(\mathbf{h}), \qquad L_S(\mathbf{h}) = \frac{1}{m}\sum_i l(\mathbf{y}_i, \mathbf{h}(\mathbf{x}_i))$$

- **Difficulties**
  - **Ill-posed**: no-uniqueness, smoothness and convexity are required for optimisation methods
  - **Under and Overfitting**:
    - $L_D(\mathbf{h})$ the "true error" (to be defined)
    - $\mathbf{h}_D$ the "best hypothesis" on $\mathcal{H}$ (to be defined)

$$\mathbf{h}_D = \underset{\mathbf{h}\in\mathcal{H}}{\operatorname{argmin}}\ L_D(\mathbf{h})$$

$$L_D(\mathbf{h}_D) \leq L_D(\hat{\mathbf{h}}_S), \qquad L_S(\hat{\mathbf{h}}_S) \leq L_S(\mathbf{h}_D)$$

  - Approximation and generalisation errors

$$L_D(\hat{\mathbf{h}}_S) = \underbrace{L_D(\mathbf{h}_D)}_{\underset{|\mathcal{H}|\to+\infty}{\to} 0} + \underbrace{\Big(L_D(\hat{\mathbf{h}}_S) - L_D(\mathbf{h}_D)\Big)}_{\leq c\sqrt{\frac{|\mathcal{H}|}{m}}}$$

Outline

1 Make-up on Physics-based Ground motion simulation
- Hybrid models
- Seismic ground motion
- The first hybrid model

2 Deep Make-up on physics-based simulation
- Supervised learning
- Unsupervised Learning
- Generative Adversarial Networks
- Adversarial Generative auto-encoders

3 Questions
- Convergence
- Stochastic gradient

Stochastic gradient

- **Assumption**: $L_S(\mathbf{h}) = L_S(\mathbf{W})$ is smooth and locally convex ($l = \frac{1}{2}\|\mathbf{y} - \mathbf{h}(\mathbf{x})\|^2$)

$$\nabla_{\mathbf{W}} L_S = \frac{1}{m} \sum_i \nabla_{\mathbf{W}} \mathbf{h}(\mathbf{x}_i) \cdot \underbrace{\nabla_{\mathbf{h}} l(\mathbf{y}_i, \mathbf{h}(\mathbf{x}_i))}_{\Delta_i = \mathbf{y}_i - \mathbf{h}(\mathbf{x}_i)}$$

$$\nabla_{\mathbf{W}} \mathbf{h}(\mathbf{x}) \Delta = \prod_{l=1}^{n} \nabla_{\mathbf{W}_l} \mathbf{h}_l(\mathbf{x}_l) \cdot \Delta_l$$

$$\nabla_{\mathbf{W}_l} \mathbf{h}_l(\mathbf{x}_l) = g_l' \left(\mathbf{W}_l \mathbf{x}_l + \mathbf{b}_l\right) \mathbf{x}_l \Delta_l$$

- **Stochastic gradient**

$$\mathbf{W}^{(j+1)} = \mathbf{W}^{(j)} + \underbrace{\alpha}_{\text{Learning rate}} \nabla_{\mathbf{W}} \mathbf{h}(\mathbf{x}_i) \Delta_i$$

Probabilistic approach
Log-Likelihood

- **Log-Likelihood** of S

$$L_{\hat{Q}}(S) = -L_S(\hat{Q}) \quad = \quad \frac{1}{m}\sum_i \ln \hat{Q}(\mathrm{Y}_i|\mathrm{X}_i) - \mathrm{cst}$$

- **Log-Likelihood Maximisation**

$$\hat{Q}_S = \underset{Q\in\mathcal{Q}}{\operatorname{argmax}}\ L_Q(S)$$

- **Generalisation error**

$$\begin{aligned} L_D(Q) &= E\left[D_{\mathrm{KL}}(P\|QP_{\mathrm{X}})\right] \\ &= E\left[-\ln Q(\mathrm{Y}|\mathrm{X})\right] - \overbrace{S(P_{\mathrm{Y}|\mathrm{X}})}^{\text{Entropy of } P_{\mathrm{Y}|\mathrm{X}}} \end{aligned}$$

  - **Consistency**

$$L_D(Q) = E\left[L_S(Q)\right]$$