IOCA 2021

MDPI

Article

A Generative Adversarial Network Based Autoencoder for Structural Health Monitoring

Giorgia Colombera [1,*], Luca Rosafalco [1], Matteo Torzoni [1], Filippo Gatti [2], Stefano Mariani [1], Andrea Manzoni [3] and Alberto Corigliano [1]

[1] Dipartimento di Ingegneria Civile ed Ambientale, Politecnico di Milano, Piazza L. da Vinci 32, 20133 - Milano (Italy)
[2] Université Paris Saclay - CNRS - CentraleSupélec, Laboratoire MSSMat UMR 8579, 3 rue Joliot Curie, 91190 - Gif sur Yvette (France)
[3] Dipartimento di Matematica, Politecnico di Milano, Piazza L. da Vinci 32, 20133 - Milano (Italy)
* Correspondence: giorgia.colombera@mail.polimi.it

Abstract: Civil structures, infrastructures and lifelines are constantly threatened by natural hazards and climate change. Structural Health Monitoring (SHM) has therefore become an active field of research in view of online structural damage detection and long term maintenance planning. In this work we propose a new SHM approach leveraging a deep Generative Adversarial Network (GAN), trained on synthetic time histories representing the structural responses of both damaged and undamaged multistory building to earthquake ground motion. In the prediction phase, the GAN generates plausible signals for different damage states, based only on undamaged recorded or simulated structural responses, thus without the need to rely upon real recordings linked to damaged conditions.

Keywords: Structural Health Monitoring; Machine Learning; Generative Adversarial Network.

Citation: Title. *Proceedings* **2021**, *1*, 0. https://doi.org/

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Submitted to *Proceedings* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/ 4.0/).

1. Introduction

Bridges, power generation systems, aircrafts, buildings and rotating machinery are only few instances of structural and mechanical systems which play an essential role in the modern society, even if the majority of them are approaching the end of their original design life [1]. Taking into account that their replacement would be unsustainable from an economic standpoint, alternative strategies for early damage detection have been actively developed so to extend the basis service life of those infrastructures. Furthermore, the advent of novel materials whose long-term behaviour is still not fully understood drives the effort for effective Structural Health Monitoring (SHM), resulting in a saving of human lives and resources [1].

SHM consists of three fundamental steps: (i) measurement, at regular intervals, of the dynamic response of the system; (ii) selection of damage-sensitive features from the acquired data; (iii) statistical analysis of those attributes to assess the current health state of the structure. To characterize the damage state of a system, the method relying on hierarchical phases, originally proposed by [2] represents the currently adopted standard. The latter prescribes several consecutive identification phases (to be tackled in order), namely: check the existence of the damage, the location of the damage, its type, extent and the system's prognosis. Damaged states are identified by comparison with a reference condition, assumed to be undamaged. The detection of the damage location relies upon a wider awareness of the structural behaviour and the way in which it is influenced by damage. This information, along with the knowledge of how the observed features are altered by different kinds of damage, allows to determine the type of damage. The last two phases require an accurate estimation of the damage mechanisms in order to classify its severity and to estimate the Remaining Useful Life (RUL).

All the steps mentioned above rely on continuous data acquisition and processing to obtain information about the current health condition of a system. In the last few years, the concept of *Digital Twin* has emerged, combining data assimilation, machine learning and physics-based numerical Simulations [1], the latter being essential to completely understand the physics of the structure and damage mechanisms. A suitable tool able to extract main dominant features from a set of data is represented by neural networks [3], especially generative models such as Generative Adversarial Networks (GANs) [4] and Variational Autoencoders (VAEs) [5].

In this paper, an application of the generative neural network RepGAN, proposed by [6], is presented in the context of SHM. Section 2 provides an overview on existing works. In Section 3, the application of RepGAN to Structural Health Monitoring is presented. In Section 4, extensive numerical results are illustrated, while Section 5 gathers some concluding remarks.

2. Related work

Generative Adversarial Networks [4] are well known due to their generative capability. Given a multidimensional random variable $\boldsymbol{X} \in (\mathbb{R}^{d_X}, \mathcal{E}_X, P_X)$[1] (whose samples are collected in the data set $\mathcal{S} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$), with probability density function $p_X(\boldsymbol{X})$, the GAN generator attempts to reproduce *synthetic* samples $\hat{\boldsymbol{x}}$, sampled according to the probability density function $p_G(\boldsymbol{X})$ as *similar* as possible to the original data, i.e. a GAN trains over data samples in order to match p_G with p_X. $\boldsymbol{G}$ maps a lower dimension manifold $(\mathbb{R}^{d_Z}, \mathcal{E}_Z, P_Z)$ (with $d_Z < d_X$ in general) into the physics space $(\mathbb{R}^{d_X}, \mathcal{E}_X, P_X)$. In doing so, $\boldsymbol{G}$ learns to pass the *critic test*, undergoing the judgement of a discriminator $D : \mathbb{R}^{d_X} \to [0, 1]$, simultaneously trained to recognize $\hat{\boldsymbol{x}}^{(i)}$ counterfeits. The adversarial training scheme relies on the following two-players Minimax game:

$$\begin{gathered} \{\boldsymbol{G}; D\} = \arg\min_{\boldsymbol{G}} \max_{D} V(D, \boldsymbol{G}) \\ V(D, \boldsymbol{G}) = \mathbb{E}_{\boldsymbol{X} \sim p_X}[\ln D(\boldsymbol{X})] + \mathbb{E}_{\boldsymbol{Z} \sim p_Z}[\ln(1 - D(\boldsymbol{G}(\boldsymbol{Z}))] \end{gathered} \tag{1}$$

In practice, $\boldsymbol{G}$ is represented by a neural network $\boldsymbol{G}_{\boldsymbol{\theta}}$ and D by a neural network $D_{\boldsymbol{\omega}}$, with trainable weights and biases $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. Moreover, $V(D, \boldsymbol{G})$ is approximated by the Empirical Risk function $L_{\mathcal{S}}(\boldsymbol{\omega}, \boldsymbol{\theta})$[2] defined as:

$$\begin{aligned} \{\boldsymbol{\theta}; \boldsymbol{\omega}\} &= \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\omega}} L_{\mathcal{S}}(\boldsymbol{\omega}, \boldsymbol{\theta}) = \\ &= \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\omega}} \frac{1}{n} \sum_{i=1}^{n} \Big(\ln D_{\boldsymbol{\omega}}(\boldsymbol{x}^{(i)}) + \ln(1 - D_{\boldsymbol{\omega}}(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i)})) \Big) \end{aligned} \tag{2}$$

with $\boldsymbol{z}^{(i)}$ sampled from a known latent space probability distribution p_Z (for instance the normal distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$). The generator $\boldsymbol{G}_{\boldsymbol{\theta}}$ induces a sampling probability $p_G(\boldsymbol{X}; \boldsymbol{\theta})$ so that, when optimized, passes the *critic test*, with D being unable to distinguish between $\boldsymbol{x}^{(i)}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z}^{(i)}\right)$ (i.e. $D(\boldsymbol{x}^{(i)}) = \frac{1}{2} = D(\boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z}^{(i)}\right))$. In other words, $\boldsymbol{x}^{(i)}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z}^{(i)}\right)$ can be associated to the value of a categorical variable C, with two possible values: class "d" (*data*) and class "g" (*generated*). $\boldsymbol{x}^{(i)}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}\left(\boldsymbol{z}^{(i)}\right)$ can be therefore sampled with the mixture probability density $p_M = \alpha\chi(C = \text{“}d\text{”}) + (1 - \alpha)\chi(C = \text{“}g\text{”})$ with χ being the indicator function and $\alpha = P(C = \text{“}d\text{”})$ [7]. The optimum solution of the Minimax game in Equation (2) induces a mixture probability distribution $\frac{1}{2}(p_{C=\text{“}d\text{”}} + p_{C=\text{“}g\text{”}})$ [4]. The saddle point of $V(D, G)$ corresponds to the minimum (with the respect to to D) of the conditional Shannon's entropy $\mathbb{S}(C|\boldsymbol{X})$ (see Appendix A). Moreover, minimizing the conditional Shannon's entropy $\mathbb{S}(C|\boldsymbol{X})$ corresponds to the maximization of the Mutual

[1] $(\mathbb{R}^{d_X}, \mathcal{E}_X, P_X)$ denotes the probabilistic space with σ-algebra $\mathcal{E}_X$ and probability measure P_X

[2] Empirical Loss depending on the data set $\mathcal{S}$

Information $I(\boldsymbol{X}, C) = \mathbb{S}(C) - \mathbb{S}(C|\boldsymbol{X})$ (see Appendix B), i.e. it corresponds to extract $\boldsymbol{X}$ samples $\boldsymbol{x}^{(i)}$ or $\hat{\boldsymbol{x}}^{(i)}$ that are indistinguishable (belonging to same class), with an uninformative mapping $\boldsymbol{X} \to C$.

GANs proved useful in various applications such as generation of artificial data for data-set augmentation, filling gaps in corrupted images and image processing. Especially, deep convolutional generative adversarial networks (DCGANs) [8] proved useful in the field of unsupervised learning. SHM could benefit from GANs as they improve the generalisation performance of models, extracting general features from data, as well as their semantics (damage state, frequency content, etc). However, the adversarial training scheme in Appendix C does not grant a bijective mapping $\boldsymbol{G}_{\boldsymbol{\theta}_Z} : \boldsymbol{Z} \to \boldsymbol{X}$ (decoder) and $\boldsymbol{F}_{\boldsymbol{\theta}_X} : \boldsymbol{X} \to \boldsymbol{Z}$ (encoder), which is crucial in order to obtain a unique representation of the data into the latent manifold. Autoencoders have been developed for image reconstruction so to learn the identity operator $\hat{\boldsymbol{x}}^{(i)} = \mathcal{I}(\boldsymbol{x}^{(i)}) = \boldsymbol{G}_{\boldsymbol{\theta}_Z} \circ \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})$. One can leverage the encoder $\boldsymbol{F}_{\boldsymbol{\theta}_X}$ representation power to sample points $\hat{\boldsymbol{z}}^{(i)} = \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})$ belonging to the latent manifold Ω_Z and the decoder $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$ to sample points $\hat{\boldsymbol{x}}^{(i)} = \boldsymbol{G}_{\boldsymbol{\theta}_Z}(\boldsymbol{z}^{(i)})$ belonging to the latent manifold Ω_X (see Equation (1)). In order to make the learning process of GANs stable across a range of data-sets and to realize higher resolution and deeper generative models, Convolutional Neural Networks (CNNs) are employed to define $\boldsymbol{F}_{\boldsymbol{\theta}_X}$, $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$ and the discriminators. $\boldsymbol{F}_{\boldsymbol{\theta}_X}$ and $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$ induce sampling probability density functions $q_{Z|X} = \frac{q_{XZ}}{p_X}$ and $p_{X|Z} = \frac{p_{XZ}}{p_Z}$ respectively. p_X is usually unknown (depending on the data-set at stake), but p_Z can be chosen *ad-hoc* (such as, for instance, $\mathcal{N}(\mathbf{0}, \mathbb{I})$) in order to get a powerful generative tool for realistic data samples $\hat{\boldsymbol{x}}^{(i)}$. A particular type of Autoencoders, called Variational Autoencoders (VAEs) was introduced by [5], consisting in a probabilistic and generative version of the standard Autoencoder, where the encoder $\boldsymbol{F}_{\boldsymbol{\theta}_X}$ infers the mean $\boldsymbol{\mu}_Z$ and variance $\boldsymbol{\sigma}_Z^2$ of the latent manifold. However, the main contribution provided by VAEs is the straightforward approach that allows to reorganize the gradient computation and reduce variance in the gradients labelled *reparametrization trick*.

Adversarial Autoencoders (AAEs) [9] employ the adversarial learning framework in Equation (1), replacing $\boldsymbol{G}_{\boldsymbol{\theta}_Z}(\boldsymbol{z}^{(i)})$ by $\boldsymbol{G}_{\boldsymbol{\theta}_Z} \circ \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})$ and adding to the adversarial GAN loss the Mean Square Loss $\|\boldsymbol{x}^{(i)} - \boldsymbol{G}_{\boldsymbol{\theta}_Z} \circ \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})\|^2$ as an optimization penalty, in order to assure a good reconstruction of the original signal. However, AAEs do not assure a bijective mapping between $(\mathbb{R}^{d_X}, \mathcal{E}_X, P_X)$ and $(\mathbb{R}^{d_Z}, \mathcal{E}_Z, P_Z)$. In order to achieve the bijection (in a probabilistic sense) between $(\boldsymbol{x}, \hat{\boldsymbol{z}})$ and $(\hat{\boldsymbol{x}}, \boldsymbol{z})$ samples, the *distance* between the joint probability distributions $q_{X\hat{Z}} = q_{\hat{Z}|X} p_X$ and $p_{\hat{X}Z} = p_{\hat{X}|Z} p_Z$ [10], with the posteriors $q_{\hat{Z}|X}$ and $p_{Z|\hat{X}}$ must be minimized. A suitable *distance* operator for probability distributions is the so called *Jensen-Shannon distance* $\mathbb{D}_{JS}(q_{X\hat{Z}} || p_{\hat{X}Z})$, defined as [10]:

$$\mathbb{D}_{JS}(q_{X\hat{Z}} || p_{\hat{X}Z}) = \frac{\mathbb{D}_{KL}(q_{X\hat{Z}} \| p_M) + \mathbb{D}_{KL}(p_{\hat{X}Z} \| p_M)}{2} = \mathbb{S}(p_M) - \mathbb{S}(\boldsymbol{X}, \boldsymbol{Z}|M) \tag{3}$$

with $\mathbb{D}_{KL}(p\|q) = \mathbb{S}(p\|q) - \mathbb{S}(p)$ being the Kullback-Leibler divergence (see Appendix B) and $p_M = \frac{q_{X\hat{Z}} + p_{\hat{X}Z}}{2}$ being the mixture probability distribution [7], i.e. the probability of extracting $(\boldsymbol{X}, \hat{\boldsymbol{Z}})$ or $(\hat{\boldsymbol{X}}, \boldsymbol{Z})$ from a mixed data set, with $\alpha = P(C = \text{“}d\text{”}) = \frac{1}{2}$ and the entropy of the mixture probability $\mathbb{S}(M) = \ln 2$. $\mathbb{D}_{JS}(q_{X\hat{Z}} || p_{\hat{X}Z})$ can be rewritten as:

$$\begin{aligned} \mathbb{D}_{JS}(q_{X\hat{Z}} || p_{\hat{X}Z}) &= \mathbb{S}(p_M) - \frac{1}{2}(\mathbb{S}(q_{X\hat{Z}}) + \mathbb{S}(p_{\hat{X}Z})) = \\ &= \mathbb{S}(\boldsymbol{X}, \boldsymbol{Z}) - \mathbb{S}(\boldsymbol{X}, \boldsymbol{Z}|M) = \mathbb{S}(M) - \mathbb{S}(M|\boldsymbol{X}, \boldsymbol{Z}) \end{aligned} \tag{4}$$

The adversarial optimization problem expressed in Equation (1) can be seen as a minimization of the Jensen-Shannon distance for $C \in \{\text{“}d\text{”}, \text{“}g\text{”}\}$:

$$\begin{aligned}\mathbb{D}_{JS}(q_{X\hat{Z}}||p_{\hat{X}Z}) + \ln 2 = -\mathbb{S}(M|\boldsymbol{X}, \boldsymbol{Z}) = \\ = \frac{1}{2}\mathbb{E}_{(\boldsymbol{X},\hat{\boldsymbol{Z}})\sim q_{XZ}}\left[D(\boldsymbol{X}, \hat{\boldsymbol{Z}})\right] + \frac{1}{2}\mathbb{E}_{(\hat{\boldsymbol{X}},\boldsymbol{Z})\sim p_{XZ}}\left[1 - D(\hat{\boldsymbol{X}}, \boldsymbol{Z})\right]\end{aligned} \tag{5}$$

that can be combined with the Autoencoder model in order to obtain the following expression [10,11]:

$$\begin{aligned}\mathbb{D}_{JS}(q_{X\hat{Z}}||p_{\hat{X}Z}) + \ln 2 = \mathbb{D}_{JS}\Big(q_{\hat{Z}|X}p_X||p_{\hat{X}|Z}p_Z\Big) + \ln 2 = \\ = \frac{1}{2}\Big[\mathbb{E}_{\boldsymbol{X}\sim p_X}\Big[D\Big(\boldsymbol{X}, \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{X})\Big)\Big] + \mathbb{E}_{\boldsymbol{Z}\sim p_Z}\Big[1 - D\Big(\boldsymbol{G}_{\boldsymbol{\theta}_Z}(\boldsymbol{Z}), \boldsymbol{Z}\Big)\Big]\Big]\end{aligned} \tag{6}$$

In this context, $\boldsymbol{F}_{\boldsymbol{\theta}_X}$ learns to map data into a disentangled latent space, generally following the normal distribution, a good reconstruction is not ensured unless the cross-entropy between $\boldsymbol{X}$ and $\boldsymbol{Z}$ is minimized too [12].
Another crucial aspect of generative models is the semantics of the latent manifold. Most of the standard GAN models trained according to Equation (1) employs a simple factored continuous input latent vector $\boldsymbol{Z}$ and does not enforce any restrictions on the way the generator treats it. The individual dimensions of $\boldsymbol{Z}$ do not correspond to semantic features of the data (uninformative latent manifolds) and $\boldsymbol{Z}$ cannot be effectively used in order to perform meaningful topological operations in the latent manifold (e.g., describing neighborhoods) and to associate meaningful labels to it. An information-theoretic extension to GANs, called InfoGAN [13] is able to learn a meaningful and disentangled representations in a completely unsupervised manner: a Gaussian noise $\boldsymbol{Z}$ is associated to a latent code $\boldsymbol{C}$ to capture the characteristic features of the data distribution (for classification purposes). As a consequence, the generator becomes $\boldsymbol{G}_{\boldsymbol{\theta}_Z}(\boldsymbol{Z}, \boldsymbol{C})$ and the corresponding probability distribution p_G, whose Mutual Information with the respect to to the latent codes $\boldsymbol{C}$, namely $I(\boldsymbol{C}, \boldsymbol{G}_{\boldsymbol{\theta}_Z}(\boldsymbol{Z}, \boldsymbol{C}))$. The latter is forced to be high, penalizing the GAN loss in Equation (1) with the variational lower bound $L_I(G,Q)$, defined by:

$$L_I(G, Q) = \mathbb{E}_{\boldsymbol{C}\sim p_C, \boldsymbol{X}\sim p_G}[\ln Q(\boldsymbol{C}|\boldsymbol{X})] + \mathbb{S}(\boldsymbol{C}) = \mathbb{E}_{\boldsymbol{X}\sim p_G}\mathbb{E}_{\boldsymbol{C}\sim p_{C|X}}\left[\ln q_{C|X}\right] + \mathbb{S}(\boldsymbol{C}) \tag{7}$$

with $q_{C|X}$ being the probability distribution approximating the real unknown posterior probability distribution $p_{C|X}$ (and represented by the neural network $\boldsymbol{Q}_Z$). $L_I(G, Q)$ can be easily approximated via Monte Carlo simulation, and maximized with the respect to to $q_{C|X}$ and p_G via reparametrization trick [13].

$$V_{\text{InfoGAN}}(D, G, Q) = V(D, G) - \lambda L_I(G, Q) \tag{8}$$

3. Methods

With the purpose of learning a semantically meaningful and disentangled representation of the SHM time-histories, we adopted in this study the architecture called RepGAN, originally proposed in [6]. RepGAN is based on an encoder-decoder structure (both represented by deep CNNs made of stacked 1D convolutional blocks), with a latent space $\boldsymbol{Z} = [\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{N}]$. $\boldsymbol{C} \in [0,1]^{d_C}$ a categorical variable representing the damage class(es), with $\boldsymbol{C} \sim p_C$ which is generally chosen as a categorical distribution over d_C classes, i.e. $p_C = \text{Cat}(d_C)$. $\boldsymbol{S} \in \mathbb{R}^{d_S}$ is a continuous variable of dimension d_S, with $\boldsymbol{S} \sim p_S$, generally $p_S = \mathcal{N}(\boldsymbol{0}, \mathbb{I})$ or the uniform distribution $p_S = \mathcal{U}(-1, 1)$. Finally, $\boldsymbol{N} \in \mathbb{R}^{d_N}$ is a random noise of d_N independent components, with $\boldsymbol{N} \sim p_N$, generally $p_N \sim \mathcal{N}(\boldsymbol{0}, \mathbb{I})$. RepGAN adopts the conceptual frameworks of VAEs and InfoGAN, combining the learning of two representations $x \to \hat{z} \to \hat{x}$ and $z \to \hat{x} \to \hat{z}$ respectively. The $x \to \hat{z} \to \hat{x}$ scheme must learn to map multiple data instances $\boldsymbol{x}^{(i)}$ into their images

(via encoder $\boldsymbol{F}_{\boldsymbol{\theta}_X}$) in a latent manifold $\hat{\boldsymbol{z}}^{(i)} = \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})$ and back into a distinct instance in data space $\hat{\boldsymbol{x}}^{(i)} = \boldsymbol{G}_{\boldsymbol{\theta}_Z} \circ \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{x}^{(i)})$ (via decoder $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$), providing satisfactory results in reconstruction. $z \to \hat{x} \to \hat{z}$ maps multiple data latent instances into the same data representation, in order to guarantee impressive generation and clustering performance. Combining the two surjective mappings, in RepGAN the two learning tasks $x \to \hat{z} \to \hat{x}$ and $z \to \hat{x} \to \hat{z}$ are performed together with shared parameters in order to obtain a bijective mapping $x \leftrightarrow z$ [3]. This ability to learn a bidirectional mapping between the input space and the latent space is achieved through a symmetric adversarial process. The Empirical Loss function can be written as:

$$\begin{aligned} L_S = {} & \mathbb{D}_{JS}\Big(p_{\hat{X}|(C,S,N)}||p_X\Big) + \mathbb{D}_{JS}\Big(q_{\hat{C}|X}||p_C\Big) + \mathbb{D}_{JS}\Big(q_{\hat{S}|X}||p_S\Big) + \mathbb{D}_{JS}\Big(q_{\hat{N}|X}||p_N\Big) \\ & - \mathbb{E}_{p_C}\Big[\mathbb{E}_{p_{\hat{X}|C}}\Big[\ln q_{\hat{C}|X}\Big]\Big] - \mathbb{E}_{p_S}\Big[\mathbb{E}_{p_{\hat{X}|S}}\Big[\ln q_{\hat{S}|X}\Big]\Big] - \mathbb{E}_{p_X}\Big[\mathbb{E}_{q_{(C,S,N)|X}}\Big[\ln p_{X|(C,S,N)}\Big]\Big] \end{aligned} \tag{9}$$

with the terms:

- $-\mathbb{E}_{p_C}\Big[\mathbb{E}_{p_{\hat{X}|C}}\Big[\ln q_{\hat{C}|X}\Big]\Big]$ minimizing the conditional entropy $\mathbb{S}(\boldsymbol{C}|\boldsymbol{X})$
- $-\mathbb{E}_{p_S}\Big[\mathbb{E}_{p_{\hat{X}|S}}\Big[\ln q_{\hat{S}|X}\Big]\Big]$ minimizing the conditional entropy $\mathbb{S}(\boldsymbol{S}|\boldsymbol{X})$

are introduced in order to constrain a deterministic and injective encoding mapping (see Appendix B). On the other hand, the term

- $-\mathbb{E}_{p_X}\Big[\mathbb{E}_{q_{(C,S,N)|X}}\Big[\ln p_{X|(C,S,N)}\Big]\Big]$

penalizes the learning scheme, in order to minimize the conditional entropy $\mathbb{S}(\boldsymbol{X}|(\boldsymbol{C},\boldsymbol{S},\boldsymbol{N}))$, i.e. in order to grant a good reconstruction.
Following the original RepGAN formulation:

- $-\mathbb{E}_{p_X}\Big[\mathbb{E}_{q_{(C,S,N)|X}}\Big[\ln p_{X|(C,S,N)}\Big]\Big]$ is enforced penalizing the L_1-norm $|\boldsymbol{X} - \boldsymbol{G}_{\boldsymbol{\theta}_Z} \circ \boldsymbol{F}_{\boldsymbol{\theta}_X}(\boldsymbol{X})|$
- $\mathbb{E}_{p_S}\Big[\mathbb{E}_{p_{\hat{X}|S}}\Big[\ln q_{\hat{S}|X}\Big]\Big]$ corresponds to the InfoGAN L_I penalty, and it is maximized via the *reparametrization trick* (structuring the $\boldsymbol{S}$ branch of the encoder-decoder structure as a VAE, see [5])

Finally, $\mathbb{E}_{p_C}\Big[\mathbb{E}_{p_{\hat{X}|C}}\Big[\ln q_{\hat{C}|X}\Big]\Big]$ is maximized in a supervised way, considering the actual class of labeled signals $\boldsymbol{x}^{(i)}$: $\boldsymbol{x}_d^{(i)}$ corresponding to a damaged structure and $\boldsymbol{x}_u^{(i)}$ to an undamaged one respectively. RepGAN provides an informative and disentangled latent space associated to the damage class $\boldsymbol{C}$. The most significant aspect of the approach is the efficiency in generating reasonable signals for different damage states only on the basis of undamaged recorded or simulated structural responses. Both generators $\boldsymbol{F}_{\boldsymbol{\theta}_X}$, $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$ and discriminators $D_{\boldsymbol{\omega}_X}$, $D_{\boldsymbol{\omega}_C}$, $D_{\boldsymbol{\omega}_S}$ and $D_{\boldsymbol{\omega}_N}$ are parametrized via 1D CNN (and strided 1D CNN), following [8]. Our RepGAN model has been designed using the Keras API, and trained employing a Nvidia Tesla K40 GPU (on the supercomputer *Ruche*, the cluster of the Mésocentre Moulon of Paris Saclay University).

4. Results and Discussion

In the following, a case study is considered in order to prove the ability of the new architecture to achieve the three fundamental tasks of semantic generation, clustering and reconstruction. The reference example is a shear building subject to an earthquake ground motion whose signals are taken from the STEAD seismic database [14]. STEAD [14] is a high-quality, large-scale, and global data set of local earthquake and non-earthquake signals recorded by seismic instruments. In this work, local earthquake wave forms (recorded at *local* distances within 350 km of earthquakes) have been considered. Seismic

[3] In practice, the training of $z \to \hat{x} \to \hat{z}$ is iterated five times more than the $x \to \hat{z} \to \hat{x}$

data are constituted by three wave forms of 60 seconds duration, recorded in east-west, north-south, and up-dip directions respectively. The structure is composed by 39 storeys. The mass and the stiffness of each floor, in undamaged conditions, are respectively $m = 625 \cdot 10^3$ kg and $k = 8.33 \cdot 10^7 \ \frac{kN}{m}$. Damage is simulated through the degradation of stiffness. In the present case, the stiffness reduction has been set equal to 50% of the above mentioned value. The structural response of the system is evaluated considering one degree-of-freedom (dof) per floor. To take into account damping effects, a Rayleigh damping model has been considered.

The following results have been obtained considering 100 signals in both undamaged and damaged conditions for a total of 200 samples, with separated training and validation data sets. Each signal is composed by 2048 time steps with dt = 0.04 s. The training process has been performed over 2000 epochs. The reconstruction capability of the proposed network has been evaluated through the Goodness-of-Fit (GoF) criteria [15] where both the fit in Envelope (EG) and the fit in Phase (FG) are measured. An example is shown in Figure 1. The values 9.17 and 9.69 respectively related to EG and PG testify the excellent reconstruction quality.

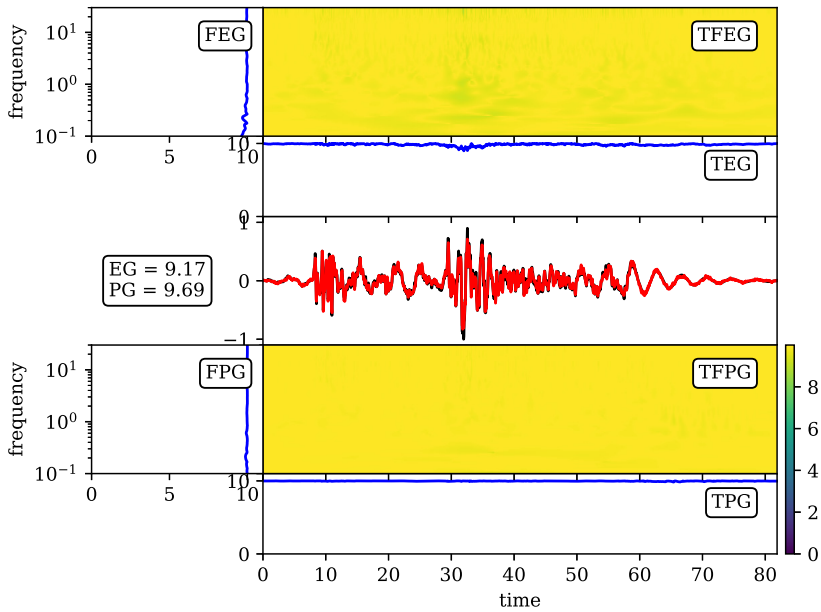

Figure 1. Time–Frequency Goodness-of-Fit criterion: the black line represents the original time-histories $\boldsymbol{x}^{(i)}$ while the red time history depicts the result of the RepGAN reconstructions $\boldsymbol{G}_Z \circ \boldsymbol{F}_X\left(\boldsymbol{x}^{(i)}\right)$. GoF is evaluated between 0 and 10: the higher the score, the better is the reconstruction. Frequency Envelope Goodness (FEG), Time–Frequency Envelope Goodness (EG), Time Envelope Goodness (TEG), Frequency Phase Goodness (FPG), Time–Frequency Phase Goodness (PG) and Time Phase Goodness (TPG).

The capability of reproducing signals for different damage scenarios can be appreciated from Figure 2 which presents the original structural response (black) and the corresponding generated one (orange) in both undamaged (left panel in Figure 2) and damaged (right panel in Figure 2) conditions. Regarding the classification capability, the classification report and the confusion matrix in Figure 3 highlight the fact that the model is able to correctly assign the damage class to the considered time histories.

5. Conclusions

In this paper, we introduce a SHM method based on a deep Generative Adversarial Network. Trained on synthetic time histories that represent the structural response of a multistory building in both damaged and undamaged conditions, the new model achieves

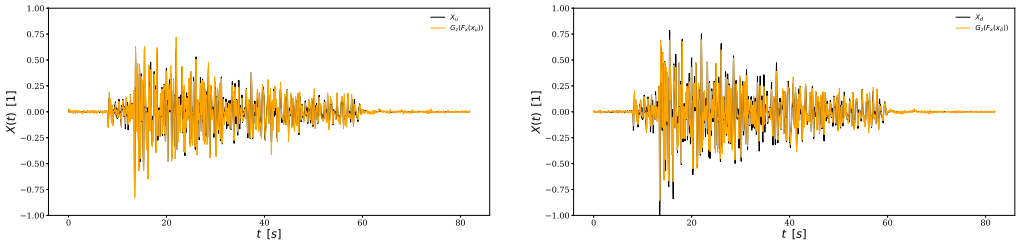

Figure 2. Examples of reconstructed signals for undamaged (left) and damaged (right) time-histories. The black lines represent the original time-histories $\boldsymbol{x}_u^{(i)}$ and $\boldsymbol{x}_d^{(i)}$ respectively. The orange time histories represent the result of the RepGAN reconstructions $\boldsymbol{G}_Z \circ \boldsymbol{F}_X\left(\boldsymbol{x}_u^{(i)}\right)$ and $\boldsymbol{G}_Z \circ \boldsymbol{F}_X\left(\boldsymbol{x}_d^{(i)}\right)$ respectively. The proposed examples represent the normalized displacement of the 1st floor of the building in object.

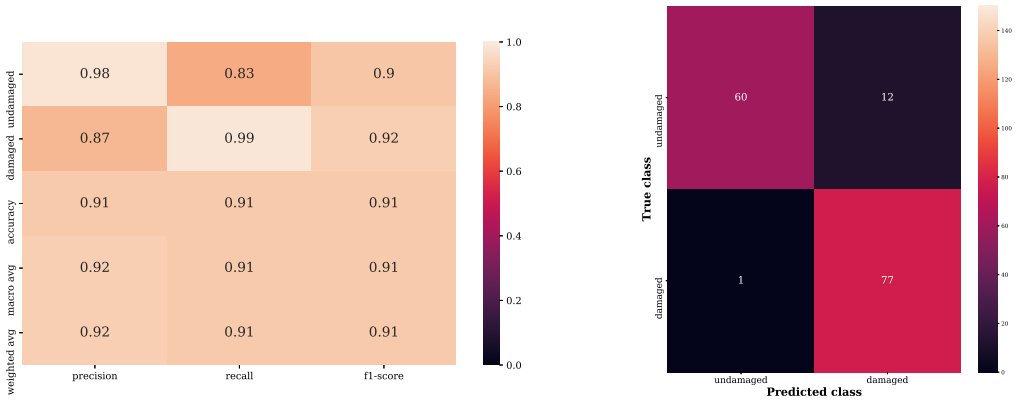

Figure 3. Evaluation of the classification ability of the model. On the left panel, precision, recall, f1-score and accuracy values are reported. A precision score of 1.0 for a class C means that every item labelled as belonging to class C does indeed belong to class C, whereas a recall of 1.0 means that every item from class C was labelled as belonging to class C. F1-score is the harmonic mean of the precision and recall. Accuracy represents the proportion of correct predictions among the total number of cases examined. On the right panel, the confusion matrix allows to visualize the performance of the model: each row of the matrix represents the instances in the actual class, while each column depicts the instances in the predicted class.

high classification accuracy (Figure 3) and satisfactory reconstruction quality (Figure 1, Figure 2), resulting in a good bidirectional mapping between the input space and the latent space. However, the major innovation of the proposed method is the ability to generate reasonable signals for different damage states, based only on undamaged recorded or simulated structural responses. As a consequence, real recordings linked to damaged conditions are not requested. In our future work, we would like to extend our approach to real-time data. We will further consider a dataset constituted by a far larger number of time histories.

Acknowledgments

The training and testing of the neural network has been performed exploiting the supercomputer resources of the Mésocentre Moulon[4], the cluster of CentraleSupélec and ENS Paris-Saclay, hosted within the Paris-Saclay University and funded by the Contrat Plan État Région (CPER).
This work has been developed thanks to the scholarship “Tesi all’estero - a.y. 2020/2021 - second call” funded by Politecnico di Milano.

[4] http://mesocentre.centralesupelec.fr

Appendix A. Shannon's entropy

- Shannon's entropy for a probability density function p_X:

$$\mathbb{S}(\boldsymbol{X}) = \mathbb{S}(p_X) = \mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\ln \frac{1}{p_X}\right] = -\mathbb{E}_{\boldsymbol{X}\sim p_X}[\ln p_X] \geq 0$$

- Conditional Shannon's entropy for X and Z:

$$\mathbb{S}(X|Z) = \mathbb{E}_{\boldsymbol{Z}\sim p_Z}[\mathbb{S}(p_{X|Z})] = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Z})\sim p_{XZ}}\left[\ln\left(\frac{1}{p_{Z|X}}\right)\right]$$

$$\mathbb{S}(X,Z) = \mathbb{S}(Z|X) - \mathbb{S}(X) = \mathbb{S}(X|Z) - \mathbb{S}(Z)$$

- Cross-entropy:

$$\mathbb{S}(p_{XZ}||q_{XZ}) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Z})\sim p_{XZ}}\left(\ln\left(\frac{1}{q_{XZ}}\right)\right) = \mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\mathbb{E}_{\tilde{\boldsymbol{Z}}\sim p_{Z|X}}\left[\ln\left(\frac{1}{q_{XZ}}\right)\right]\right]$$

- Given a data set of identically independent distributed (i.i.d.) samples $\mathcal{S} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$, the true yet unknown probability p_X of extracting an instance $\boldsymbol{x}^{(i)}$ can be approximated by the likelihood $p_{\theta_X}\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$, whose entropy is

$$\mathbb{S}(p_{\theta_X}) = -\ln p_{\theta_X}\left(\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}\right) = \sum_{i}^{N} \ln p_{\theta_X}\left(\boldsymbol{x}^{(i)}\right)$$

Appendix B. Kullback-Leibler divergence

- Kullback-Liebler divergence (non-symmetric):

$$\mathbb{D}_{KL}(p_{XZ}\|q_{XZ}) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Z})\sim p_{XZ}}\left[\ln\left(\frac{p_{XZ}}{q_{XZ}}\right)\right] = -\mathbb{S}(p_{XZ}) + \mathbb{S}(p_{XZ}||q_{XZ}) \leq \mathbb{S}(p_{XZ}||q_{XZ})$$

$$\mathbb{D}_{KL}(p_{XZ}\|q_{XZ}) + \mathbb{S}(X) = \underbrace{-\mathbb{S}(X|Z)}_{\mathbb{E}_{\boldsymbol{X}\sim p_X}[\mathbb{S}(p_{Z|X})]} + \mathbb{S}(p_{XZ}||q_{XZ}) \leq \mathbb{S}(p_{XZ}||q_{XZ})$$

$$\mathbb{D}_{KL}(p_{XZ}\|q_{XZ}) + \mathbb{S}(X) \leq \mathbb{D}_{KL}(p_{XZ}\|q_{XZ})$$

- Mutual Information between $\boldsymbol{X}$ and $\boldsymbol{X}|\boldsymbol{Z}$:

$$I(\boldsymbol{X},\boldsymbol{Z}) = \mathbb{S}(\boldsymbol{X}) - \mathbb{S}(\boldsymbol{X}|\boldsymbol{Z}) \geq 0$$

If $p_{X|Z} = p_X$ ($(\boldsymbol{X},\boldsymbol{Z})$ are independent) then $I(\boldsymbol{X},\boldsymbol{Z}) = 0$. If $p_{X|Z} = \delta(\boldsymbol{Z} - \boldsymbol{f}(\boldsymbol{X}))$ with $\boldsymbol{f}$ deterministic, then $I(\boldsymbol{X},\boldsymbol{Z}) = \max_{(\boldsymbol{X},\boldsymbol{Z})} I(\boldsymbol{X},\boldsymbol{Z}) = \mathbb{S}(\boldsymbol{X})$

- $\mathbb{S}(\boldsymbol{Z}|\boldsymbol{X}) = -\mathbb{E}_{\boldsymbol{Z}\sim p_Z}\left[\mathbb{E}_{\boldsymbol{X}\sim p_{X|Z}}\left[\ln p_{Z|X}\right]\right] =$
$= -\mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\mathbb{E}_{\boldsymbol{Z}\sim p_{Z|X}}\left[\ln \frac{p_{Z|X}}{q_{Z|X}}\right]\right] - \mathbb{E}_{\boldsymbol{Z}\sim p_Z}\left[\mathbb{E}_{\boldsymbol{X}\sim p_{X|Z}}\left[\ln q_{X|Z}\right]\right] =$
$= -\mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\mathbb{D}_{KL}\left(p_{Z|X}\|q_{Z|X}\right)\right] - \mathbb{E}_{\boldsymbol{Z}\sim p_Z}\left[\mathbb{E}_{\boldsymbol{X}\sim p_{X|Z}}\left[\ln q_{X|Z}\right]\right] \leq$
$\leq -\mathbb{E}_{\boldsymbol{Z}\sim p_Z}\left[\mathbb{E}_{\boldsymbol{X}\sim p_{X|Z}}\left[\ln q_{X|Z}\right]\right]$

Appendix C. Generative Adversarial Networks (GAN)

- Given $\boldsymbol{X}$ belonging to the probabilistic space $(\Omega_X, \mathcal{E}_X, P_X)$ with class $C \in \{\text{“}d\text{”}, \text{“}g\text{”}\}$ (“d” corresponding to *data* and “g” to *generated*, and a *discriminator* $D : \Omega_X \to [0,1]$ acting as an *expert/critic*:
  - $P(C = \text{“}d\text{”}) = \alpha; \quad P(C = \text{“}g\text{”}) = 1 - \alpha$

- $P(C = \text{“}d\text{”}|\boldsymbol{x}^{(i)}) = D(\boldsymbol{x}^{(i)})$
- $P(C = \text{“}d\text{”}|\boldsymbol{x}^{(i)}) = 1 - D(G(\boldsymbol{z}^{(i)}))$

$$\mathbb{S}(C|\boldsymbol{X}) = -\mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\mathbb{E}_{C\sim p_{C|\boldsymbol{X}}} \ln\left(p_{C|\boldsymbol{X}}\right)\right] = -\mathbb{E}_{C\sim p_C}\left[\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}|C}}\left[\ln\left(p_{C|\boldsymbol{X}}\right)\right]\right]$$

$$\mathbb{S}(C|\boldsymbol{X}) = -\alpha\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}|C=\text{“}d\text{”}}}\left[\ln\left(p_{C=\text{“}d\text{”}|\boldsymbol{X}}\right)\right] - (1-\alpha)\mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}|C=\text{“}g\text{”}}}\left[\ln\left(p_{C=\text{“}g\text{”}|\boldsymbol{X}}\right)\right]$$
$$\mathbb{S}(C|\boldsymbol{X}) = -\alpha\mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\ln(D(\boldsymbol{X}))\right] - (1-\alpha)\mathbb{E}_{\boldsymbol{Z}\sim p_Z}\ln(1 - D(\boldsymbol{G}(\boldsymbol{Z})))$$

For tuneable conditional probability distributions $D_{\boldsymbol{\omega}}$:

$$\max I(\boldsymbol{X}, C) \leq \mathbb{S}(C) + \max -\mathbb{S}(C|\boldsymbol{X}) = \mathbb{S}(C) + \min \mathbb{S}(C|\boldsymbol{X})$$

$$\max I(\boldsymbol{X}, C) \leq \mathbb{S}(C) + \min_G \max_D \alpha\mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\ln(D(\boldsymbol{X}))\right](1-\alpha)\mathbb{E}_{\boldsymbol{Z}\sim p_Z}\ln(1 - D(\boldsymbol{G}(\boldsymbol{Z})))$$

Thus minimizing $\mathbb{S} + \min_G \max_D \mathbb{S}(C|\boldsymbol{X})$ represents an upper bound for the Mutual Information between C and $\boldsymbol{X}$, which is maximized by maximizing $-\mathbb{S}(C|\boldsymbol{X})$. For an optimum training, D must not be able to discriminate between $\boldsymbol{x}^{(i)}$ and $\hat{\boldsymbol{x}}^{(i)}$, therefore $\alpha = \frac{1}{2}$.

Appendix D. Standard Autoencoder [5]

In the standard Autoencoder formulations [16,17], $\boldsymbol{F}$ and $\boldsymbol{G}$ are trained by maximizing $I(\boldsymbol{X}, \boldsymbol{Z})$, namely:

$$\{\boldsymbol{F}, \boldsymbol{G}\} = \arg\max_{\boldsymbol{F},\boldsymbol{G}} I(\boldsymbol{X}, \boldsymbol{Z}) = \arg\min_{\boldsymbol{F},\boldsymbol{G}} H(\boldsymbol{X}|\boldsymbol{Z}) = \arg\min_{\boldsymbol{F},\boldsymbol{G}} \mathbb{E}_{\boldsymbol{X}\sim p_X}\left[\mathbb{E}_{\boldsymbol{Z}\sim q_{Z|X}}\left[\ln\left(\frac{1}{p_{X|Z}}\right)\right]\right] \tag{A1}$$

If the encoder and decoder are parametrized as neural networks, respectively as $\boldsymbol{F}_{\boldsymbol{\theta}_X}$ and $\boldsymbol{G}_{\boldsymbol{\theta}_Z}$, the AE loss can be approximated by the Empircal Loss:

$$\{\boldsymbol{\theta}_X, \boldsymbol{\theta}_Z\} = \arg\max_{\boldsymbol{\theta}_X,\boldsymbol{\theta}_Z} \sum_{i=1}^{N}\left[\ln\left(p_{X|Z}\left(\boldsymbol{x}^{(i)}|\boldsymbol{Z} = \boldsymbol{F}_{\theta_X}\left(\boldsymbol{x}^{(i)}\right)\right)\right)\right] \tag{A2}$$

Given the fact that the Gaussian distribution has maximum entropy relative to all probability distributions covering the entire real line, the Empirical Loss in Equation (A2) can be maximized by the Empirical Loss with $p_{X|Z} = \mathcal{N}\left(\boldsymbol{G}_{\theta_Z}(\boldsymbol{Z}), (\sigma^2)\mathbb{I}\right)$:

$$\{\boldsymbol{\theta}_X, \boldsymbol{\theta}_Z\} = \arg\max_{\boldsymbol{\theta}_X,\boldsymbol{\theta}_Z} \sum_{i=1}^{N} \frac{1}{2\sigma^2}\|\boldsymbol{x}^{(i)} - \boldsymbol{G}_{\theta_Z} \circ \boldsymbol{F}_{\theta_X}\left(\boldsymbol{x}^{(i)}\right)\|^2 + \frac{d_Z}{2}\ln\left(2\pi\sigma^2\right) \tag{A3}$$

References

1. C. R. Farrar; K. Worden. *Structural health monitoring: a machine learning perspective*; Wiley: Oxford, U.K., 2013. https://doi.org/10.1002/9781118443118
2. A. Rytter. Vibrational Based Inspection of Civil Engineering Structures. Ph.D. Thesis defended publicly at the University of Aalborg, April 20, 1993.
3. C. M. Bishop. *Neural Networks for Pattern Recognition*; Oxford University Press, Inc: USA, 1995.
4. I.J. Goodfellow; J. Pouget-Abadie; M. Mirza; B. Xu; D. Warde-Farley; S. Ozair; A. Courville; Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014; Montreal; Canada. https://doi.org/10.1145/3422622
5. D. P. Kingma; M. Welling. Auto-Encoding Variational Bayes. arXiv:1312.6114, 2013.
6. Zhou, Y., Gu, K., Huang, T. (2019, July). Unsupervised Representation Adversarial Learning Network: from Reconstruction to Generation. 2019 International Joint Conference

on Neural Networks (IJCNN). 2019 International Joint Conference on Neural Networks (IJCNN). https://doi.org/10.1109/ijcnn.2019.8852395

7. Lindsay, Bruce G. Mixture Models: Theory, Geometry and Applications. In NSF-CBMS Regional Conference Series in Probability and Statistics, i–163, 1995.
8. A. Radford; L. Metz, Luke; S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434 2015.
9. Adversarial Autoencoders. ArXiv Preprint ArXiv:1511.05644, 2015. Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial Autoencoders. In International Conference on Learning Representations, 2016. http://arxiv.org/abs/1511.05644.
10. Dumoulin, Vincent, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially Learned Inference, 2016. arXiv:1606.00704v3
11. J. Donahue; P. Krähenbühl; T. Darrell. Adversarial Feature Learning. arXiv:1605.09782, 2017.
12. Li, Chunyuan, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards Understanding Adversarial Learning for Joint Distribution Matching. In Advances in Neural Information Processing Systems, 5495–5503, 2017. arXiv:1709.01215
13. X. Chen; Y. Duan; R. Houthooft; J. Schulman; I. Sutskever; P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, Curran Associates Inc.: Red Hook, NY, USA. arXiv:1606.03657 2016.
14. Mousavi, S. M., Sheng, Y., Zhu, W., Beroza, G. C. (2019). STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI. IEEE Access, 7, 179464–179476. https://doi.org/10.1109/access.2019.2947848
15. M. Kristekova, J. Kristek, P. Moczo, Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals, Geophys. J. Int. 178 (2) (2009) 813–825, http://dx.doi.org/10.1111/j.1365-246X.2009.04177.x.
16. Vincent, Pascal, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In Proceedings of the 25th International Conference on Machine Learning, 1096–1103. ACM, 2008. http://icml2008.cs.helsinki.fi/papers/592.pdf.
17. Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. Journal of Machine Learning Research 11, no. Dec (2010): 3371–3408.