Hybrid physics-based/data-based seismic ground motion generator of a site

Gottfried Jacquet

To cite this version:

Gottfried Jacquet. Hybrid physics-based/data-based seismic ground motion generator of a site. Civil Engineering. Université Paris-Saclay, 2024. English. NNT : 2024UPAST035 . tel-04654168

HAL Id: tel-04654168
https://theses.hal.science/tel-04654168
Submitted on 19 Jul 2024

Hybrid physics-based/data-based seismic ground motion generator of a site

Generation hybride de la réponse sismique de site par simulation et apprentissage machine

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 579
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)
Spécialité de doctorat: Génie civil
Graduate School : Sciences de l'ingénierie et des systèmes.
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche **LMPS**
Laboratoire de Mécanique Paris-Saclay
(Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS),
sous la direction de **Didier Clouteau**, Professeur des Universités
et le co- encadrement de **Filippo Gatti**, Maître de Conférences

Thèse soutenue à Paris-Saclay, le 23 avril 2024, par

Gottfried JACQUET

Composition du jury

Membres du jury avec voix délibérative

M. Fernando LOPEZ-CABALLERO Professeur, CentraleSupélec	Président
M. Maarten ARNST Professeur, Université Libre de Liège	Rapporteur
M. Dimitris PITILAKIS Professor, Aristotle University of Thessaloniki	Rapporteur
Mme Wassila OUERDANE Maître de Conférences HDR CentraleSupélec	Examinatrice
M. Jean-Francois SEMBLAT Professeur, ENSTA Paris	Examinateur

THESE DE DOCTORAT

NNT : 2024UPAST035

université
PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et
énergétiques, matériaux
et géosciences (SMEMAG)

Title: Hybrid physics-based/data-based seismic ground motion generator of a site

Keywords: Earthquake simulation, Machine Learning, Signal-to-Signal Translation, Generative Adversarial Networks

Abstract: Accurately estimating the seismic response following an earthquake can save lives. However, limited computational resources and poorly characterized and unknown variability in geology and seismotectonic context pose significant challenges for simulations at the scale of a city or region. This thesis proposes a new approach combining adversarial learning methods and physics-based simulations to overcome these limitations, based on the SeismoALICE framework (F. GATTI and D. CLOUTEAU: "Towards blending Physics-Based numerical simulations and seismic databases using Generative Adversarial Network," CMAME 2020). Because of the random fluctuations in the mechanical properties of the geological medium, numerical simulations can only give results for low frequencies (LF) down to 5 or even 10 Hz. The design frequency for civil engineering structures and equipment, on the other hand, reaches 40 Hz. This thesis aims to simulate seismic signals with a higher frequency range [0 - 30 Hz] using knowledge of low-frequency signals and a database of recorded signals. To this end, we are developing an encoder and decoder adapted to seismic signals using a Conformer variant of attention techniques to capture the long-duration correlations present in accelerograms. The discriminator, which ensures that simulated signals resemble recorded signals, has been the subject of extensive development, enabling the encoder and decoder to be optimized using a min-max technique at the heart of adversarial machine learning methods. To force signal reconstruction, we adapt Focal Frequency Loss (FFL) and Hyper-Spherical Loss (HSL), which are more efficient for this data type, to time series. We then complement the LF signals up to 30 Hz by exploring different generation cases, one-to-one mapping, and one-to-many mapping to assess the plausibility of the reconstructions in the database. Five methods were developed: Signal-to-Signal Translation, SeismoALICE with shared latent space, SeismoALICE with factorized latent space, BicycleGAN for time series, and Multi-Modal Signal Translation. Their performance was evaluated using Kristeková's Goodness-of-Fit. By manipulating the hidden variables, we proved that it is possible to divide the information into two groups of variables with Gaussian distributions, one for low frequencies and the other for high frequencies. This interpretability made it possible to manipulate the latent space and control the one-to-many mapping. The models, trained on 128,000 seismic signals from the Stanford Earthquake Database (STEAD), demonstrated their performance, with prediction qualities ranging from good to excellent. Finally, their effectiveness was demonstrated by application to the 2019 Le Teil earthquake (in the Ardèche region of Auvergne-Rhone-Alpes, France). This work paves the way for more accurate and efficient prediction of seismic signals by seamlessly integrating physics-based knowledge and machine learning.

Titre: Génération hybride de réponses sismiques par simulation et apprentissage machine

Mots clés: Apprentissage automatique, translation de signal à signal, tremblement de terre, onde sismique, sismogramme.

Résumé: L'estimation précise de la réponse sismique suite à un tremblement de terre permet de sauver des vies. Toutefois, la limitation des ressources informatiques et la variabilité méconnue et mal caractérisée de la géologie et du contexte sismotectonique posent des défis significatifs pour les simulations à l'échelle d'une ville ou d'une région. Cette thèse propose une nouvelle approche combinant les méthodes d'apprentissage adverse (adversarial) et les simulations basées sur la physique pour surmonter ces limitations, en s'appuyant sur le cadre SeismoALICE, (F. GATTI et D. CLOUTEAU: "Towards blending Physics-Based numerical simulations and seismic databases using Generative Adversarial Network", CMAME 2020). En raison des fluctuations aléatoires des propriétés mécaniques du milieu géologiqu, les simulations numériques ne peuvent donner des résultats que pour les basses fréquences (BF) jusqu'à 5 voire 10 Hz. La fréquence de conception des structures et des équipements en génie civil atteint en revanche 40 Hz. Cette thèse vise à simuler des signaux sismiques plus riches en fréquences [0 – 30 Hz] à partir de la connaissance des signaux à basses fréquences et d'une base de données de signaux enregistrés. Dans ce but, nous développons un encodeur et un décodeur adaptés aux signaux sismiques utilisant une variante des techniques d'attention, nommée Conformer, pour capturer les corrélations de longue durée présentes dans les accélérogrammes. Le discriminateur, assurant que les signaux simulés ressemblent à des signaux enregistrés, a fait l'objet d'un développement poussé, permettant d'optimiser l'encodeur et le décodeur par le biais d'une technique de min-max au cœur des méthodes adverses d'apprentissage machine. Pour forcer a reconstruction des signaux, nous adaptons aux séries temporelles la Focal Frequency Loss (FFL) et la Hyper-Spherical Loss (HSL), qui sont plus performantes pour ce type de données. Ensuite, nous complétons les signaux BF jusqu'à 30 Hz en explorant différents cas de génération : mapping one-to-one et mapping one-to-many pour évaluer la plausibilité des reconstructions de la base de données. Cinq méthodes ont été élaborées : Signal-to-Signal Translation, SeismoALICE with shared latent space, SeismoALICE with factorized latent space, BicycleGAN for time series et Multi-Modal Signal Translation. Leur performance a été évaluée avec le Goodness-of-Fit de Kristeková. Nous avons prouvé en manipulant les variables cachées qu'il est possible de diviser l'information en deux groupes de variables de distributions Gaussiennes, l'un pour les basses fréquences et l'autre pour les hautes fréquences. Cette interprétabilité a permis de manipuler l'espace latent et de contrôler le mapping one-to-many. Les modèles, entraînés sur 128 000 signaux sismiques de la base de données des séismes de Stanford (STEAD), démontrent leur performance avec des qualités de prédiction allant de bonnes à excellentes. Finalement, leur efficacité a été démontrée par une application au séisme du Teil de 2019 (en Ardèche dans la région Auvergne-Rhone-Alpes, France). Ce travail ouvre la voie à une prédiction plus précise et plus efficace des signaux sismiques en intégrant de manière transparente les connaissances basées sur la physique et l'apprentissage machine.

I dedicate this thesis to:

my mother, Mrs Marie Elide JANVIER JACQUET

my father, Mr. René JACQUET

my brother, Dr Geoffroy JACQUET

"The love of family is life's greatest gift."

In Memory of

The 300,000 lives lost in the devastating earthquake that occurred in Haiti on January 12, 2010.

Contents

Nomenclature

- μ, mean
- σ, standard-variation
- $\mathcal{N}(0, \mathbf{I})$, gaussian distribution of mean 0 and standard deviation of 1
- $\mathbf{x}$, Physic Based Simulation acceleration
- $\hat{\mathbf{x}}$, reconstruction of Physic Based acceleration
- $\tilde{\mathbf{x}}$, hyrbrid generation of Physic Based acceleration
- $\mathbf{y}$, Ground motion or Broadband acceleration
- $\hat{\mathbf{y}}$, reconstrcution of Broadband acceleration
- $\tilde{\mathbf{y}}$, hybrid generation of Broadband acceleration
- F_x, Encoder dedicated to Physic Based Simulation Acceleration
- G_x, Decoder that outputs Physic Based Simulation acceleration
- F_y, Encoder dedicated to Broadband Acceleration
- G_y, Decoder that outputs Broadband Acceleration
- F_{xy}, Encoder with two branch
- $\mathbf{z}_x$, Latent value of Physic Based Simulation Acceleration
- $\mathbf{z}_y$, Latent value of Broadband acceleration
- $\mathbf{z}_{xx}$, Latent value from restriction to specific part of Physic-Based simulation
- $\mathbf{z}_{yy}$, Latent value from restriction to specific part of Broadband
- $\mathbf{z}_{xy}$, Latent values from common part, with PBS as input
- $\mathbf{z}_{yx}$, Latent values from common part, with Ground motion as input
- $\mathbf{c}_x$, content part coming from PBS as input
- $\mathbf{c}_y$, content part coming from Ground motion
- $\mathbf{s}_x$, style coming from PBS
- $\mathbf{s}_y$, style coming from Ground motion

Chapter 1

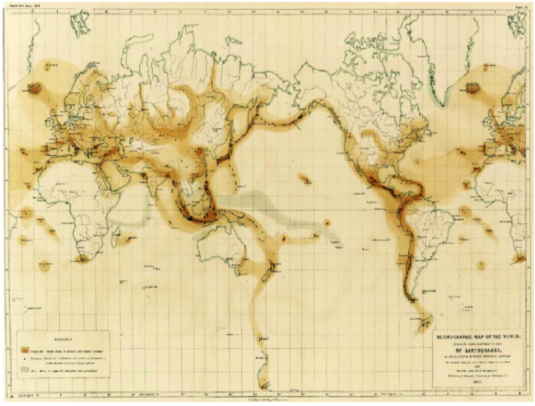
Introduction

> *"Train yourself, arm yourself with science to the teeth[..]."*
>
> — Cheikh Anta Diop, Conférence of Niamey (Niger), 1984

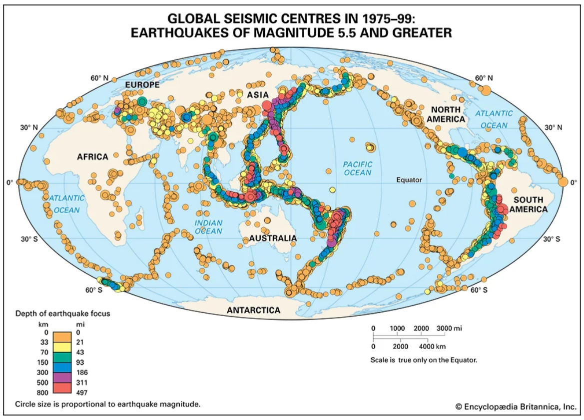
1.1 General Context

Historically attributed to divinity across various civilizations, Earthquakes underwent a transformative shift in the scientific domain around 1800. Probably after the Lisbon disaster, scientific interest in seismic phenomena emerged with the advent of experimental seismology, pioneered by the Irish engineer Robert Mallet between 1830 and 1850. Mallet's work used the earthquake intensity to produce the first seismicity map for Mediterranean regions in 1857, which remains a fundamental reference in contemporary seismological studies (see Figure 1.1a).

At the beginning of the 20th century, in-depth research into earthquakes reached a turning point. An increasingly scientific one replaced a qualitative approach to the phenomenon. Alexis Perrey and Fernand de Montessus de Ballor were the first to participate in the systematic recording of seismic activity worldwide. A significant step forward was subsequently made in the study



(a) World map of the major earthquakes, rendered by Mallet in 1858. This map shows earthquake intensity measures used to determine magnitude.[1, 2, 3]



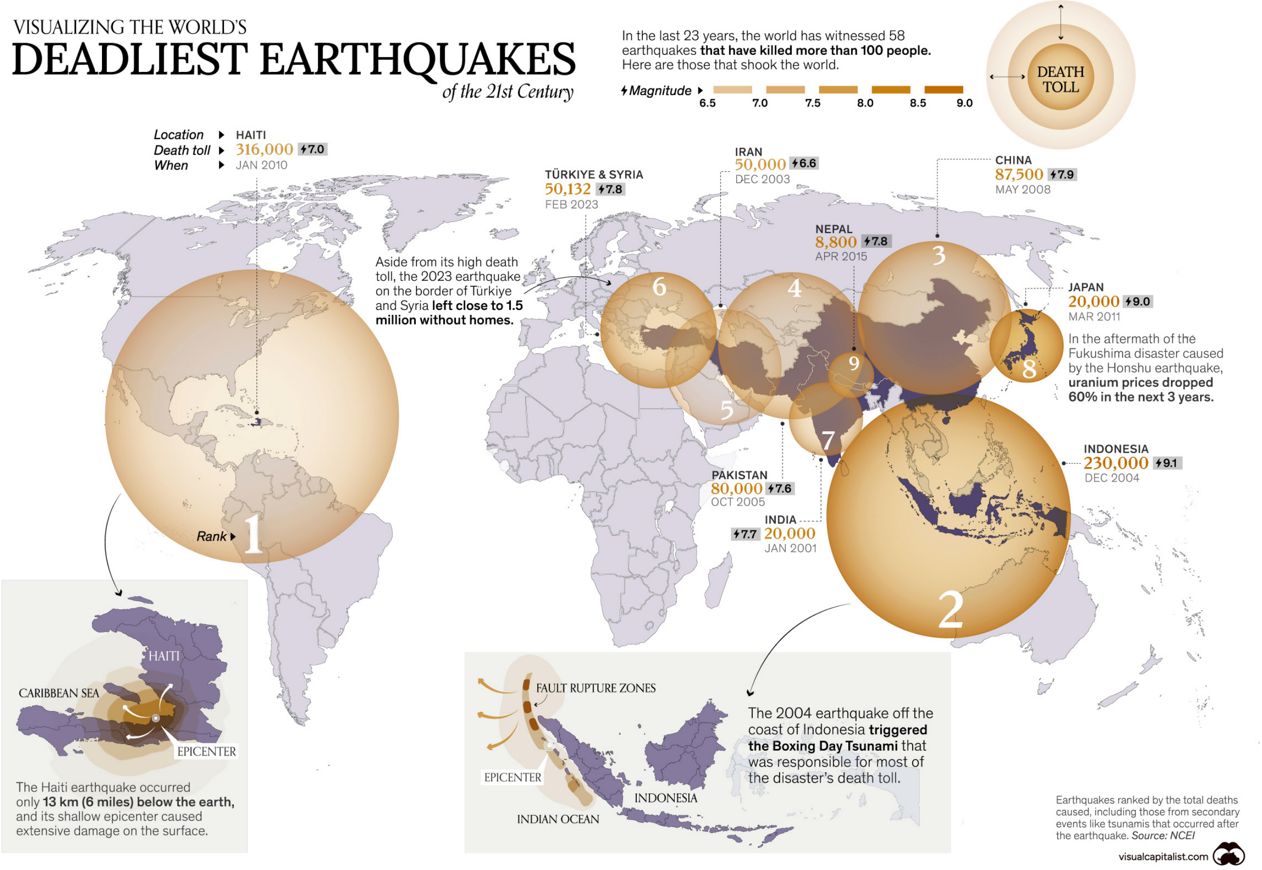
(b) An example of modern world earthquake map. Source: ©Britannica (`https://www.britannica.com/science/earthquake-geology`) of seismic waves. Richard Dixon Oldham helped to advance the field by distinguishing between different types of seismic waves. Richard Dixon's work marked the transition from descriptive seismology to an in-depth study of these geological phenomena using empirical methods.

The Earth's crust's mechanical properties can be thought of as random processes varying across different scales, from millimeters to continental scales, to be precise. This aspect implies that the inverse problem of identifying these properties based on the available measurements is naturally ill-posed, and it suffers from an intrinsic curse of dimensionality [4, 5]: the finer the geological model, the more significant the amount of data required to calibrate it, in order to obtain accurate and reliable numerical predictions. Therefore, a methodical approach based on sophisticated mathematical models and advanced numerical simulations is essential to effectively understand and reproduce the earthquake phenomenon and reach scientifically robust conclusions. The quantity of interest encompasses the acceleration time histories at the surface or other derived intensity measures (peak ground motion values, earthquake duration, among others). The attention is devoted to identifying the earthquake focal mechanism and the complex interactions of the propagation wave field with the geological structure.

A three-dimensional (3-D) wave equation in highly heterogeneous non-linear geomaterials can generically model the earthquake phenomenon. With that respect, in recent years, the credibility of numerical simulation has significantly improved [6]. These scientific achievements now make it possible to explain more convincingly an observed earthquake's strong ground motion and the corresponding recordings at the surface, both in urban areas and close to critical infrastructures. Numerical simulations, based on a solid theoretical foundation and fed with an increasing amount of seismological data, provide more accurate and realistic results, which is of vital importance for assessing the seismic risk and for planning measures to mitigate potential damage in densely populated urban areas, to help to strengthen the resilience in the face of natural disasters of seismic origin.

Moreover, earthquake-prone areas are progressively being characterized worldwide due to the critical implication of having densely populated areas exposed to this natural threat and to human and economic losses that follow. One base Modern seismic risk management on identifying and analyzing areas where high tectonic activity is detected [7, 8]. As output, the so-called *shake maps* provide a preliminary assessment of the regions likely to be affected by seismic events, which is essential for planning risk mitigation measures and for implementing emergency plans in the event of an earthquake [9], [10].

In fact, among various natural disasters, earthquakes are one of the most consequential and lethal catastrophes. Despite the achievements in scientific understanding and technological advancement, recent earthquakes resulted in enormous economic and human losses. In terms of damage to noteworthy lifelines and infrastructure, the 2011 Tōhoku earthquake and tsunami [11] caused an overall loss of \$360 billion in property damage. The 2008 Sichuan earthquake [12] caused \$150 billion loss, whereas the Turkey-Syria earthquakes of in 2023 [13] reached a loss of \$109 billion in damage. However, the real impact of earthquakes becomes most apparent when we consider the human cost, which goes beyond just economic loss (see Figure 1.2). In 2004, the Sumatra earthquake caused a tsunami that killed 227,898 people and displaced 1,126,900 individuals across 14 countries in South Asia and East Africa [14]. Similarly, the 2010 Haïti earthquake has killed between 100,000 and 316,000 lives [15], a poignant reminder of the irreplaceable cost of seismic disasters. In light of these devastating events, it becomes clear that earthquakes pose a unique and unparalleled threat, requiring ongoing efforts to mitigate their impact and protect vulnerable communities.



In the last 23 years, the world has witnessed 58 earthquakes **that have killed more than 100 people.** Here are those that shook the world.

Aside from its high death toll, the 2023 earthquake on the border of Türkiye and Syria **left close to 1.5 million without homes.**

In the aftermath of the Fukushima disaster caused by the Honshu earthquake, **uranium prices dropped 60% in the next 3 years.**

The Haiti earthquake occurred only **13 km (6 miles) below the earth,** and its shallow epicenter caused extensive damage on the surface.

The 2004 earthquake off the coast of Indonesia **triggered the Boxing Day Tsunami** that was responsible for most of the disaster's death toll.

Earthquakes ranked by the total deaths caused, including those from secondary events like tsunamis that occurred after the earthquake. *Source: NCEI*

Figure 1.2: Summary of the deadliest earthquake in the world of the 21st century. Source: geographical.co.uk

1.2 Main challenges in engineering seismology

The study of complex earthquake phenomena requires the ability to predict with a high degree of reliability the behavior of active faults and the impact of the generated seismic wave on the infrastructure over time. To achieve this, it is essential to consider the natural variability inherent to the poor knowledge of the underlying geological structures. However, dealing with this variability is often a major challenge, as it is the result of multiple interdependent factors at different scales, dealing with the stress state accumulated on the fault discontinuity due to tectonic motion, with the frictional properties of the fault's asperities, with the complex geological interfaces that scatter the wave field generated once the fault slips, the non-linear behavior of shallow soil layers, the interaction with the structures at the foundation level, as schematically shown Figure 1.3.

As part of a risk analysis, it is crucial to characterize each factor individually, which requires considerable effort to understand their complex interactions. Natural variability in geophysical phenomena arises from various sources, such as geomechanical soil properties, geochemical processes, tectonic stresses, and hydrological conditions. One can categorize such phenomena into two distinct groups: deterministic and non-deterministic.

Deterministic phenomena refer to events or occurrences that exhibit predictable and reproducible behavior. Well-established physical laws form the basis of such behavior. Appropriate mathematical models and numerical simulations can confidently predict the evolution of a system. On the other hand, non-deterministic phenomena are intrinsically unpredictable, sub-

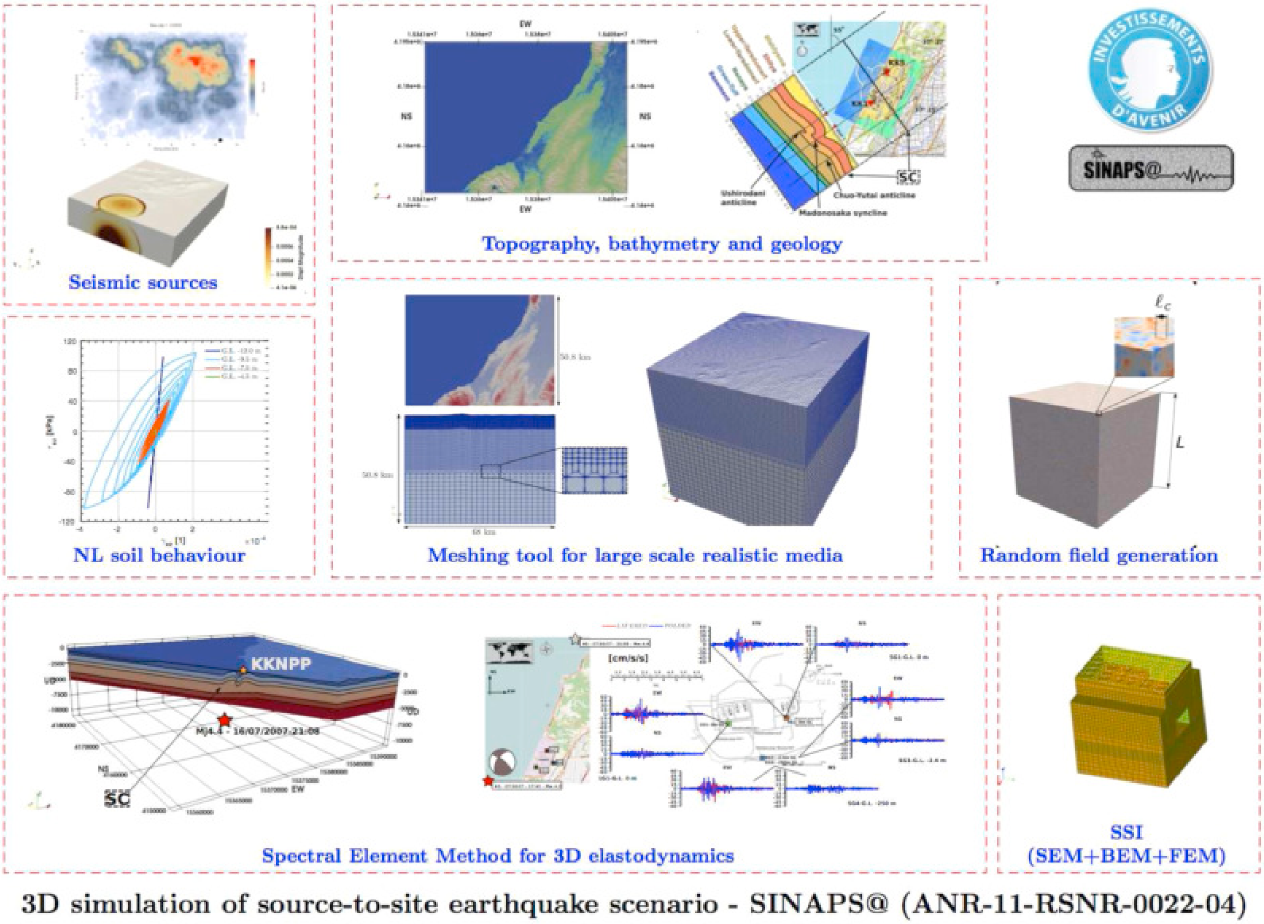
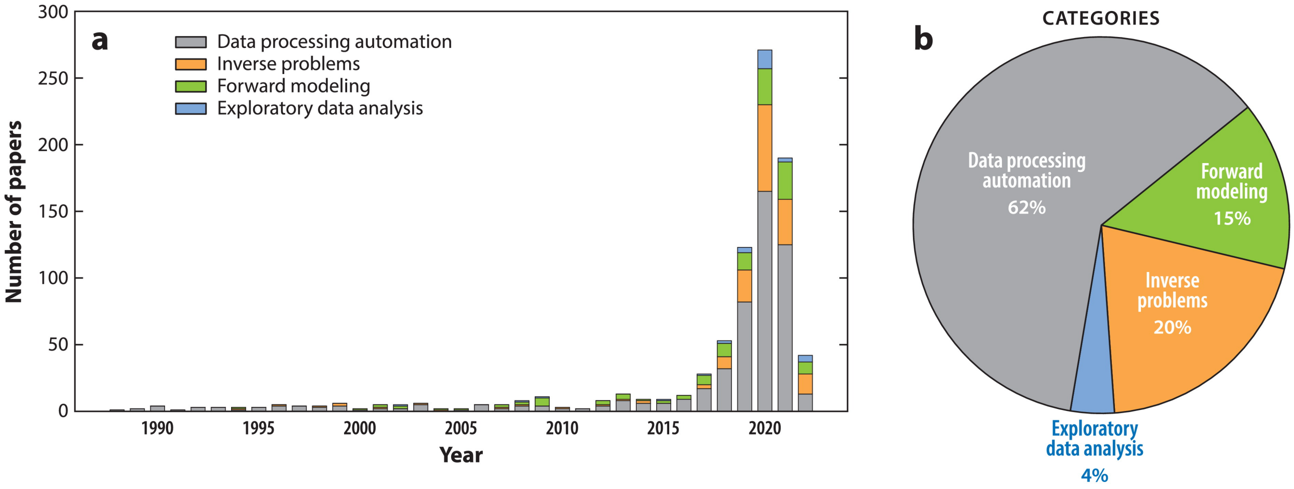


Figure 1.3: Schematic description of the multi-tool platform developed within the SINAPS@ project to reproduce realistic source-to-structure seismic scenarios. Source: Gatti *et al.*, 2018 [16]

ject to random influences or complex mechanisms that are difficult to model accurately. Their behavior is often subject to stochastic variations, making their prediction more uncertain. Modeling this variability is essential to properly understanding the risks associated with potential failures and assessing the vulnerability of exposed infrastructures and populations.
To this end, engineering seismology is increasingly making use of digital twins of regions of the Earth, i.e., numerical models that are sufficiently reliable to accurately predict the response of complex natural systems and sufficiently flexible to be continuously updated and re-calibrated according to the evolving amount of recorded measurements (for instance from geological studies, recorded seismograms, satellite images, etc.). The availability of increasingly advanced computational methods and resources enables scientists to make accurate physics-based earthquake predictions, advancing digital twin technology. (for instance, the finite element, finite different, and spectral element methods). High-performance computing on large parallel supercomputers and cloud computing allows processing massive quantities of geological and geophysical data and solving large systems of partial differential equations under various parameters' combinations in a Monte Carlo framework. Digital twins, as virtual replicas of real systems, offer the ability to simulate and analyze realistic scenarios in real-time, providing crucial information for informed decision-making.
However, due to the significant epistemic and random uncertainty associated with modeling natural phenomena, conceiving and operating a seismic digital twin is arduous. It is crucial to



Mousavi SM, Beroza GC. 2023
Annu. Rev. Earth Planet. Sci. 51:105–29

Figure 1.4: The histograms present the interest in machine learning in the seismic fields during the last decade: in the x-axis are the years, in ordinate the number of publication papers. One presents four categories of fields of investigation here. That we can see also in pie chart [18]

consider various factors of uncertainty while conducting geological risk analysis. This practice enables us to evaluate the possible outcomes of strong ground motion events accurately. As a result, we can make informed decisions regarding infrastructure development, natural resource management, and safeguarding the population in the affected areas. The use of advanced probabilistic modelling [17] and numerical simulation techniques are promising ways of assessing uncertainties and improving the accuracy of predictions in geological risk analysis, but it can be cumbersome. The geological properties of the subsoil can vary significantly, and the prospecting methods available, although expensive, often do not allow accurate extrapolation over large scales. In a nutshell, the complexity of earthquake characterization highlights the importance of developing advanced analysis methods, mainly using probabilistic approaches, to assess and model the uncertainties associated with geological and seismic parameters. Advances in numerical modeling, stochastic simulations, and geophysical data integration will help improve our understanding of complex geological phenomena and better assess seismic risks in various geological contexts.
In this framework, data-driven enrichment of seismic models is a fast-growing area of research [18].

1.3 Deep learning techniques applied to engineering seismology

Recent technological advances have seen the emergence of artificial intelligence (AI), a complex concept whose usefulness far exceeds the capabilities of traditional algorithms in specific tasks (see Figure 1.4).

This discipline has made its mark in various scientific fields, opening up new possibilities in areas where traditional research has reached its limits. Artificial intelligence demonstrates its strength from its ability to generalize from the data and research guidelines provided. Artificial intelligence can tackle intricate and complex problems, particularly those involving a large amount of data and high variability of distributions, for which it becomes complicated

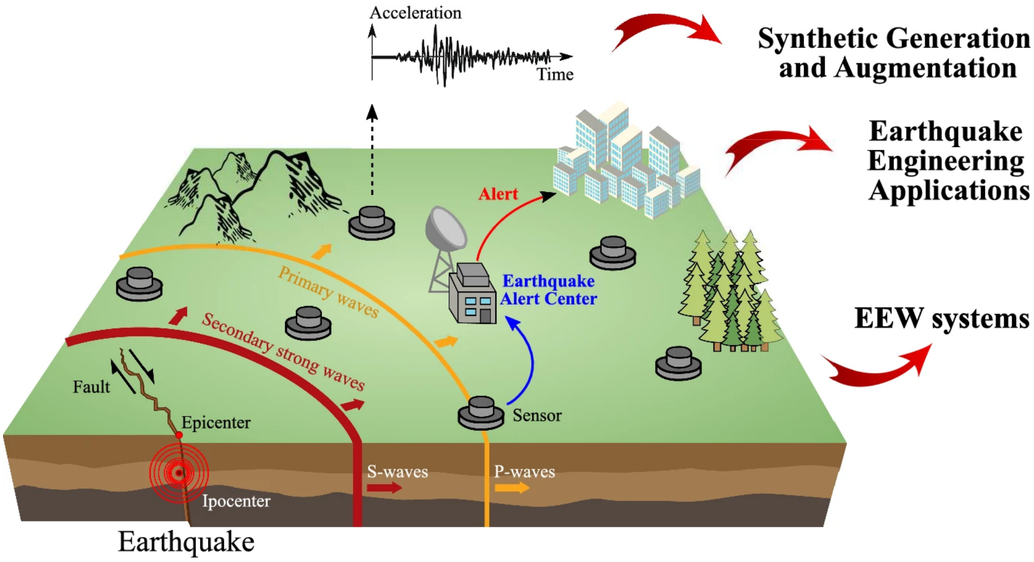


Figure 1.5: Main research activities in earthquake engineering involving machine learning techniques. Source [19]

to manipulate only a few variables to reproduce the phenomena. Such technology is suitable for dealing with complexity and improving many fields, such as medicine, finance, materials science, and geology. Thanks to those capacities, neural networks can often be employed to identify patterns that are not apparent to humans, perform tasks autonomously with great accuracy, and outperform handcrafted and assessed methods. For example, one can train a neural network model to recognize images, predict trends, perform research parameters, create predictive models, and make complex decisions in dynamic environments, particularly for earthquake warnings.

Figure 1.5 shows the statistics of the recent use of machine learning in earthquake science.

Deep learning is particularly relevant as numerical simulation models struggle to provide satisfactory accuracy in accurately predicting the seismic response at the Earth's surface as a function of data from a given accelerogram. Models of seismic wave propagation from the fault plane to the Earth's surface are limited to a frequency of 5 Hz, which hinders the field of study and potential applications in civil engineering despite advances in High-Performance Computing (HPC)., See Figure 1.6.

The potential of IA is promising in overcoming limitations and enriching numerical approaches for seismic modeling. By exploiting the considerable database of historical seismic and the associated metadata, such algorithms provide a novel way to identify the relationship between earthquake characteristics and their resulting impact on the surface area. In this sense, the prediction of such phenomena might be significantly improved by the Machine Learning approaches. The wave propagation mechanism would be better understood by countering uncertainty quantification relying on the variability of this natural phenomenon. In addition to that, IA integration in seismic modeling offers some advantages in the civil engineering field. This latter field could be more accurate in characterizing the seismic load on infrastructures and buildings. Then, by combining diverse information from historical seism and geotechnical information, IA could enhance models to mitigate earthquake impact and reinforce the design of civil engineering constructions.Figure 1.7.

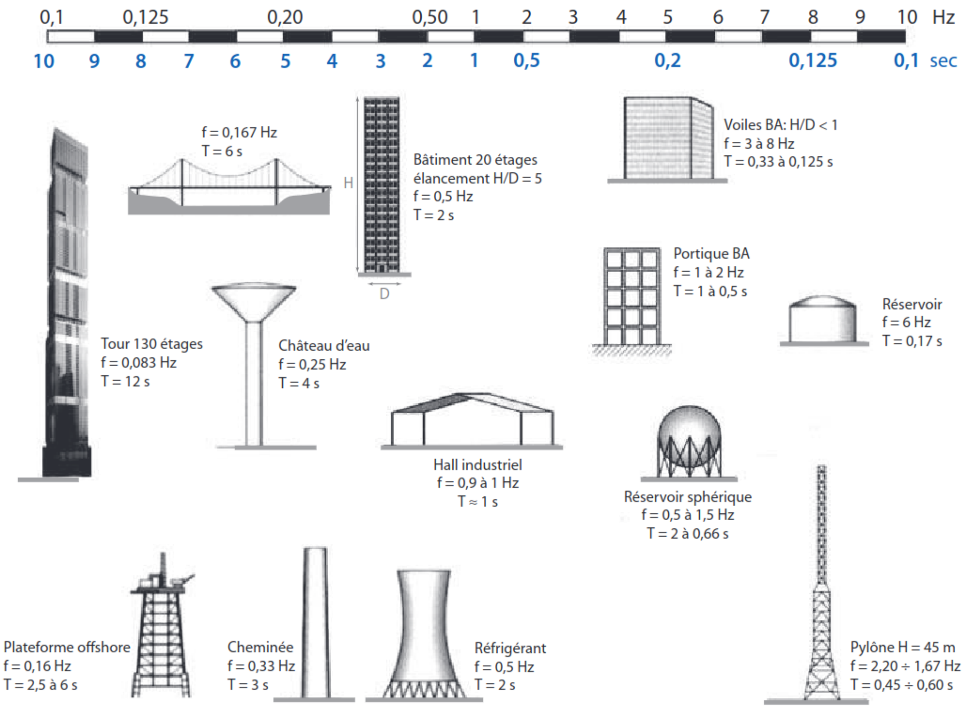


Figure 1.6: Construction frequency domain. Source Davidovici *et al.*, 2016 [20]

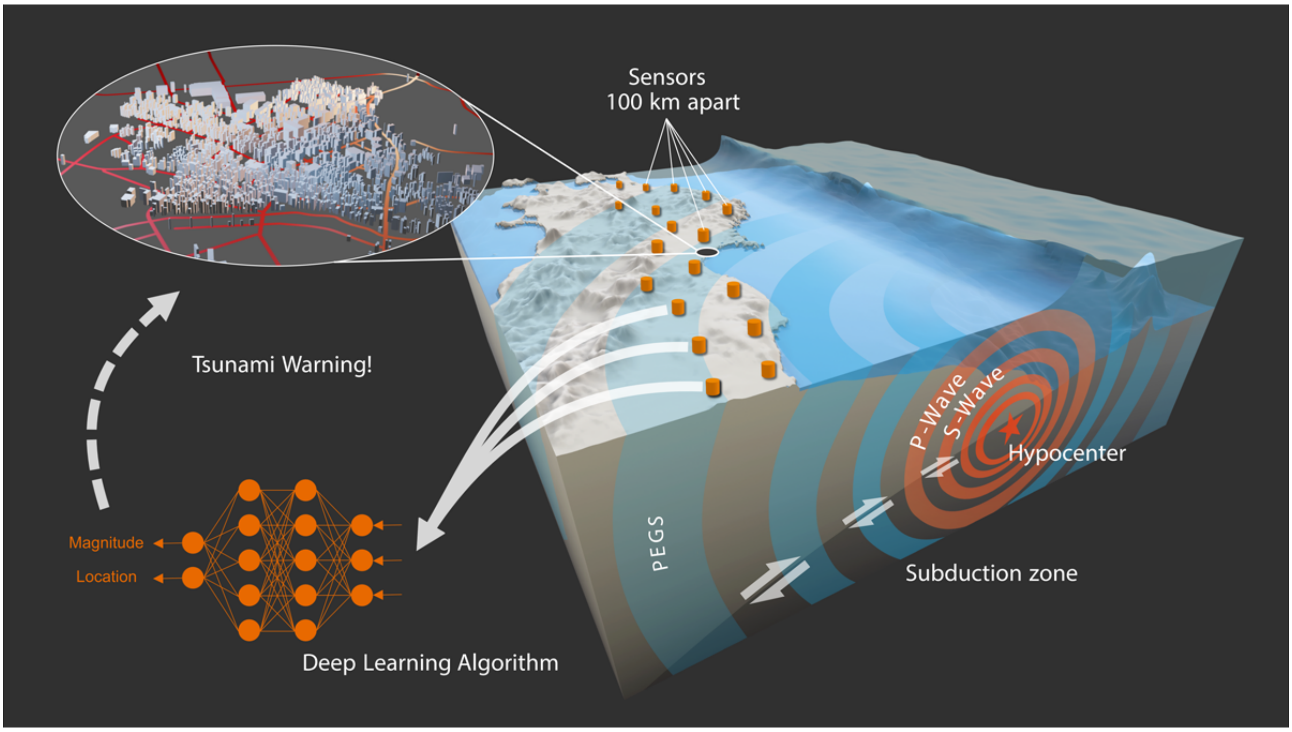


Figure 1.7: Digital twins, source Institut de recherche pour le développement (`www.humanite.fr/sciences/tsunamis/`)

1.4 Thesis Objective

It is legitimate to question the possibility of improving the results of numerical simulations by solving the so-called "super-resolution" problem. In other words, enriching the less accurate low-frequency results of numerical simulations by incorporating high-frequency information is very interesting. This will make the resulting hybrid signals closer to the recorded accelerograms. This approach, therefore, aims to fill in the gaps and inaccuracies in the numerical simulation models. (see Figure 1.8).

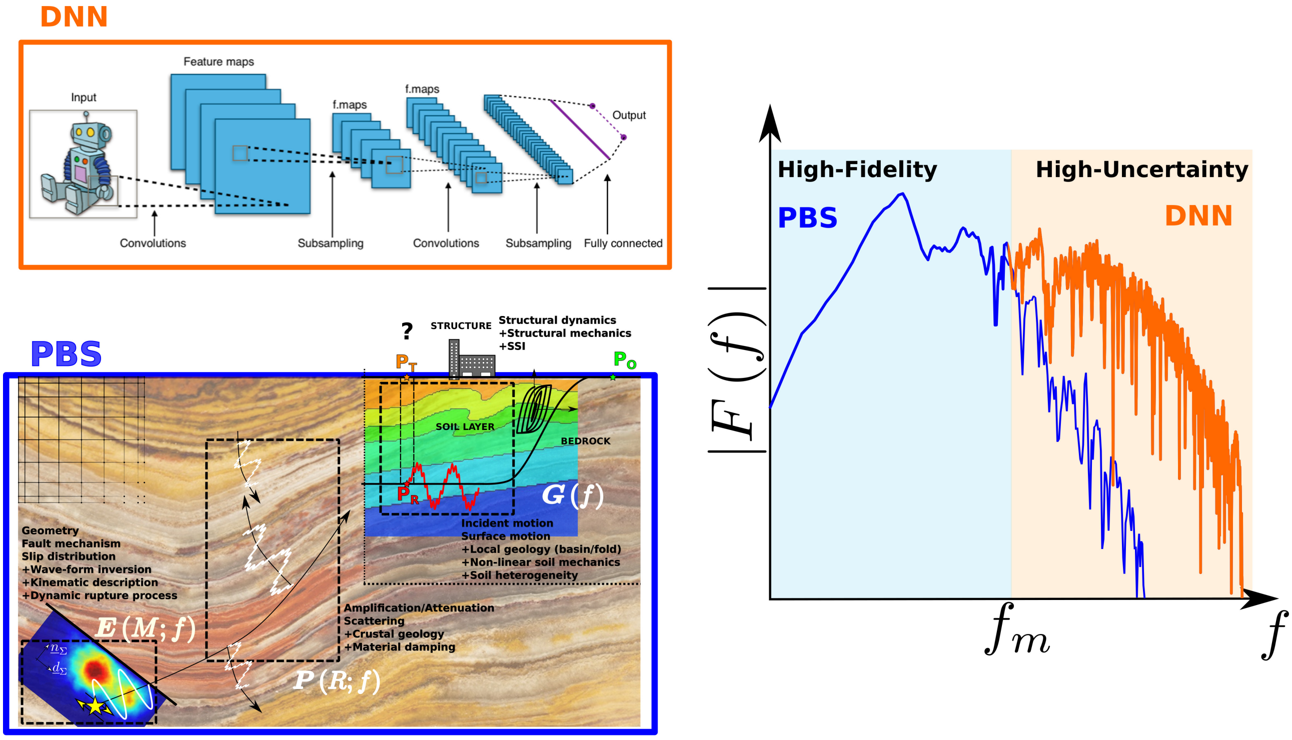


Figure 1.8: Hybrid broad-band earthquake ground motion generator. Courtesy of Gatti *et al.*, 2020 [21]

The numerical simulation is conditioned by the same low-frequency band ([0 - 1Hz]) or recorded seismogram ([0 -40Hz]). This task is crucial: Multi-modal super-resolution leverages physics-based numerical simulations independently of the maximum frequency they resolve, adding realistic high-frequency content and different broad-band realizations that can nourish structural dynamics models and vulnerability studies. This achievement would represent a paradigm shift since scientists can realize numerical simulations valid in a low-frequency band of significant confidence in input data (geology, active faults, site effects). This numerical simulation would be easier to validate against the poor amount of historical recordings in a low-frequency band. Finally, the intrinsic undisclosed mapping from low to high-frequency content, learned via AI from the real data, would generate several broad-band hybrid synthetics within a seismic hazard probabilistic framework. Therefore, a potentially infinite number of synthetic signals can be generated without having to repeat the simulation calculations (for example, 20 or 30 Hz, to be consistent with the frequency band of the natural dynamic response of civil infrastructures). This multimodal approach makes it possible to take account of the uncertainty inherent in numerical simulations, which varies according to the inputs to the problem. By generating a set of probable accelerations rather than a single value, we can better represent the variability of expected seismic results and provide complete information for decision-making in engineering, urban planning, and risk management.

1.5 How to read this document

Suppose you are an expert in Machine Learning and computer vision. In that case, you can skip most of Chapter 2, which essentially summarizes the historical evolution of machine learning techniques in seismology and Earth science (Section 2.1) and describes machine learning techniques for dimensionality reduction (Section 2.2), sample reconstruction and generation (Section 2.3) the neural network architectures suited for time histories (CNN, attention mechanisms, transformer, conformer, etc., see Section 2.4) and largely used in this work, the adopted metrics, inspired by seismology (Section 2.6) and the database of recorded earthquake time-histories employed in this thesis to train neural network metamodels (see Section 2.7). It is worth noting that subsection 2.1.7 highlights one major challenge tackled by the present thesis: the generation of synthetic earthquake seismograms with machine learning. This subsection contextualizes the whole manuscript, along with subsection 2.3.4, which introduces adversarial learning inference (ALI), adopted as a generative framework in this thesis. In this context, subsection 2.5 presents a wide range of loss functions, adopted alongside ALI, to improve the generation task. Basic machine learning concepts presented in Chapter 2 might be helpful for readers with no prior deep knowledge of modern deep learning architectures and training schemes. However, basic concepts such as back-propagation and optimizing algorithms are not reported. For further insights on such basic machine learning topics, the inexpert reader can refer to the book by Murphy, 2022 [22].

Chapter 3 outlines the preliminary results obtained to render realistic earthquake seismograms based on numerical simulations based on a widespread neural network architecture called Pix2Pix. The latter has been developed for signal-to-signal translation problems and trained in a supervised way on data pairs to generate high-quality super-resolution time series from low-resolution ones. This chapter is technical for non-expert readers in machine learning applied to time series.

In Chapter 5, we examine semi-supervised learning and its use in multi-model translation problems, utilizing BicycleGAN technology. This approach generates possible outcomes while accounting for the inherent uncertainties in the data and models. As a result, we can more fully and realistically represent diverse plausible realizations of the translation problem. This chapter, once again, is rather intricate for non-expert readers because several technical implementation details are explained to explain the advantages and disadvantages of such a strategy.

In Chapter 4, we employ unsupervised learning using ALICE and a novel version of the seminal work by Gatti and Clouteau, 2020 [21], called SeismoALICE, which originated the present research work exposed in this thesis. This innovative approach uses an unsupervised AI system to enhance the accuracy of the super-resolution task tackled by SeismoALICE. With its ability to extract complex seismic features and patterns from non-annotated data, SeismoALICE provides a practical solution to generate multiple realistic ground motion realizations in a 0-30 Hz frequency band based on the same low-frequency (0-1 Hz) physics-based simulations, embodying the minimum knowledge about the physics of the earthquake, amid considerable uncertainty on the geological and seismological features (3D underground topography, characteristics of the active faults). This chapter widens the horizons of earthquake hazard assessment in a broad-band frequency range, compatible with the modal characteristics of above-ground structures and infrastructures. It, therefore, provides multiple realistic and physically informed input seismic signals to conceive earthquake-resistant structures.

Finally, Chapter 6 is devoted to the case-study application of such machine learning techniques to assess its limits in earthquake hazard assessment. The techniques described in the previous chapters are applied to simulated time histories for the $M_W 4.9$ Le Teil earthquake that struck southeastern France in late 2019. The realism of each proposed solution is tested, adopting seismology-informed metrics and with the perspective of rendering multiple broad-band earthquake response realizations, thanks to a pre-trained neural network metamodel capable of mapping low-frequency portions of the ground motion spectra into its high-frequency counterparts.
Conclusions and future perspectives are reported in Chapter 7.

Chapter 2

State of the Art

> *"Workers are lazy people who don't know it"*
>
> — Jacques Roumain, Les Fantoches, 1937

2.1 On the Use of ML in Engineering Seismology

Over the last decade, several seismology and earthquake engineering branches have widely adopted machine learning, AI techniques, and algorithms. In the following, a non-exhaustive *excursus* of the mean findings is summarized and categorized per specific task accomplished.

2.1.1 Seismic event discrimination

Since then, numerous studies have been carried out in seismology for the discrimination of events [23, 24] to discover the exact nature of seismic occurrences. Researchers examined the seismograms recorded by a large number of stations. The recorded histories are observed in detail to extract meaningful information about the source itself (earthquake, induced seismicity, explosions, chemical explosions) and their focused mechanisms. This involves three basic principles: distinguishing between earthquake shock waves and explosions, differentiating one category of volcanic earthquakes from another, and assessing mining-related earthquakes. This latter is based on traditional methods and time-consuming [25]. However, the integration of machine learning tools has dramatically improved this task, as can be seen below [18] :

(i) **Explosions and earthquakes discrimination**: The interest in distinguishing earthquakes from chemical or quarry explosions started during The Cold War. This research field skyrocketed after the implementation of neural network detection in explosions. Notably, Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN) make a breakthrough with the classical approach based on fully connected layers [26], the latter being too heavy to be trained and prone to over-fitting. Convolutional layers, mainly, have been developed for classification because of their equivariant nature (translation insensitive). They represent the natural choice to classify different seismic events based on recorded seismograms [27, 28]. CNN outperforms standard event discrimination algorithms and helps correct human error even for low signal-to-noise ratio (SNR) data (see Figure 2.1) [18].

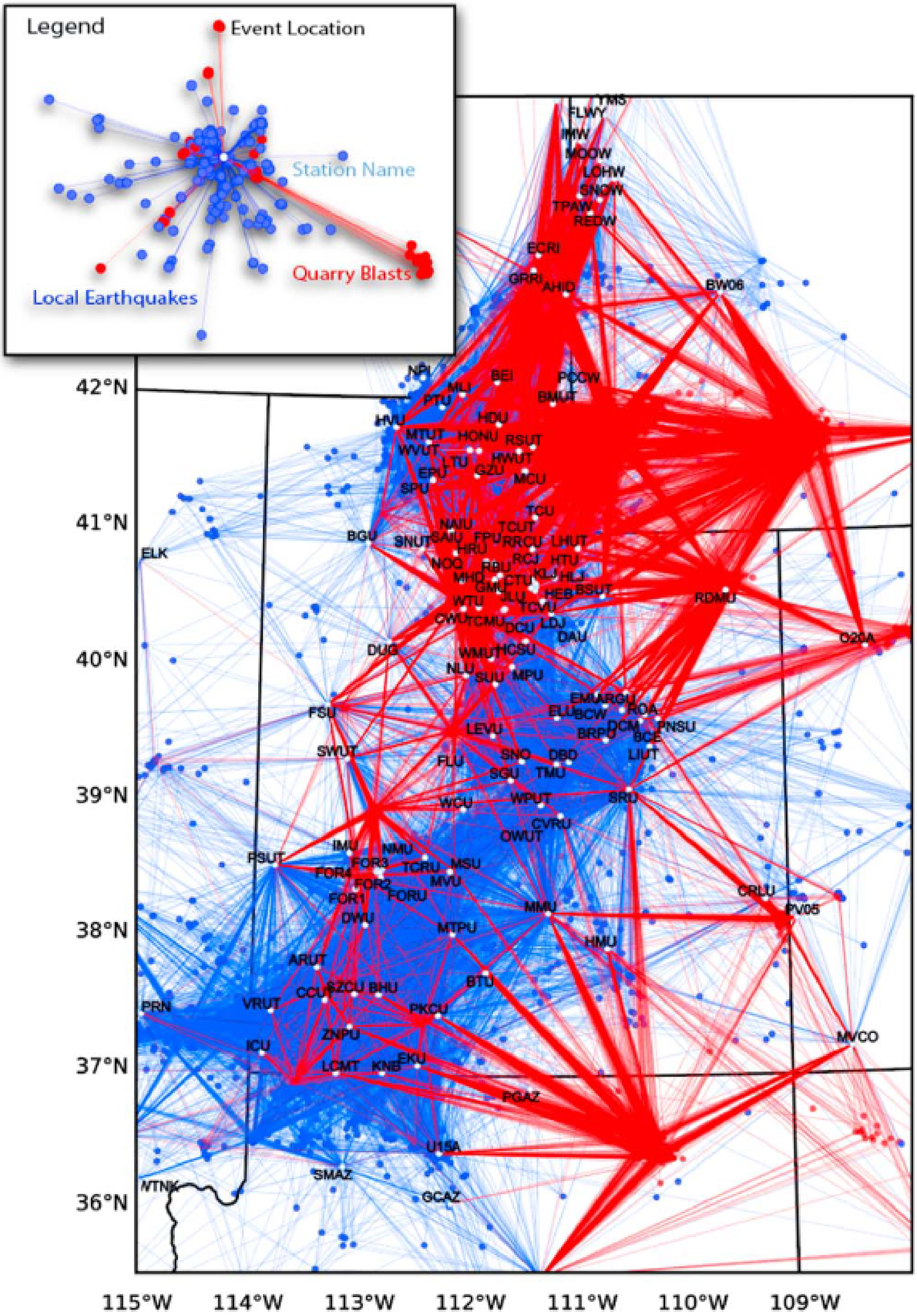


Figure 2.1: Map of events (circles) and source-receiver paths (lines) from University of Utah Seismograph Stations for quarry blasts (red) and local earthquakes (blue). Receivers (white circles) are labeled by station name. Courtesy of [27].

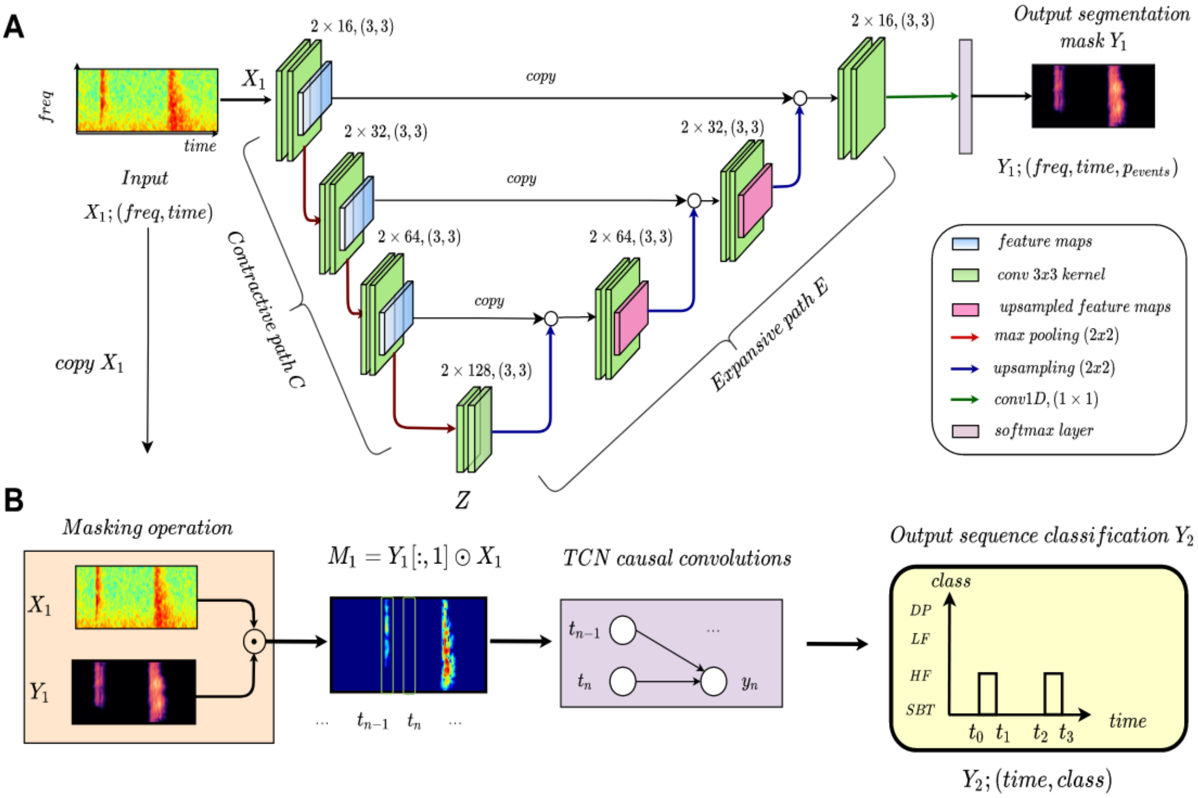


Figure 2.2: Diagram of the proposed hybrid architecture to perform seismic event detection (A) and continuous sequence classification (B) of seismo-volcanic events [30]

(ii) **Volcano-seismic events**: Volcanic eruptions can cause ground motion. In volcanology, the main problem with event detection is distinguishing magmatic events as long-period events from volcano-tectonic earthquakes, volcano tremors, and explosive earthquakes [29]. We used a neural network to differentiate different types of volcanic earthquakes, including volcano-tectonic earthquakes, long-period events, volcano-tremors, and explosive earthquakes. [30]. Despite the lack of enough labeled data for machine learning in this area [31], the latter still makes remarkable progress. Using a two-dimensional representation of time-frequency domain spectrograms has advanced image classification significantly, as shown in Figure 2.2. The introduction of a modern approach such as Wavelet Scattering Transform, depicted by features such as representation invariance, significant information content, and robust stability, has convincingly demonstrated the ongoing improvement in the accuracy of classification of Volcano Events; this improvement is demonstrated by how it was performed on Llaima volcano events in Chile where they achieved 99% accuracy score on supervised way.

(iii) **Other seismic events**: Transformers have improved seismic event classification in computer vision and language modeling. Transfer learning and data augmentation techniques have made the model's fine-tuning easier and improved result accuracy. This approach also enabled Mousavi *et al.*, 2020 [32] to come up with an unsupervised deep learning framework for feature learning and dimensionality reduction that distinguishes between local seismicity from tele seismicity waveform. Furthermore, it demonstrates a performance akin to the supervised models built on large labeled datasets, opening up possibilities for much more complicated classification tasks, such as identifying underground mining microearthquake records. Besides, Peng *et al.*, 2020 [33] described a methodology that could differentiate blast events from background noise as well as ore extraction signals, mechanical distortions, and electromagnetic interference, indicating how versatile these advanced techniques can be used in event identification tasks [34].

Machine Learning unequivocally demonstrated its effectiveness, surpassing all prior handcrafted techniques and traditional formulations.

2.1.2 Earthquake Signal Detection

Modern seismometers can detect weak ground movements, often associated with low amplitude ground motion recorded at the free surface. Approximately 90% of the time, the recorded data corresponds to ambient noise or other non-seismic phenomena. The effort to discern the distinctive seismic signature within this diverse data set is referred to as *earthquake signal detection.* Setting aside missing events and misidentifying noise as an earthquake is challenging. Traditionally, there are two main categories of conventional methods. One category design is the characteristic functions method: the Short-term average over the long-term average (STA/LTA), which is the most popular. The other category represents the similarity search (Fingerprint and Similarity thresholding, template matching, etc...). This task is enhanced by machine learning by performing binary classification of each sample point as noise or signal either along the long duration or a large number of stations. The STA/LTA times ratio method was one of the first initial approaches adapted to seismic event detection in the literature. Remarkably, this method exhibited a large robustness, leading it to major success. Employing machine learning-based detectors has since emerged as a superior alternative, surpassing conventional STA/LTA techniques, as shown in [33], adopting a moving window spectrogram as input. Over the past decades, various machine-learning detectors have been explored and incorporated. For this purpose, we could enumerate : support vector machine [35] (SVM), hidden Markov model (HMM) [36], CNN and RNN Detection (CRED) [37], multi-features fusion networks [38], capsule neural networks (CapsPhase) [39] and attentive models [40]. An example of automated event detection if provided by Lara *et al.*, 2021 [41], as shown in Figure 2.3. The use of this automated technique is very essential in detecting micro-earthquakes. An active volcano will produce many micro-earthquakes, thus making it a suitable example to explain the need for quick understanding and avoidance of these eruptions. These authors and their associates can gain valuable insights into upcoming eruptions using this method, including clustering and periodicity. This points out an increase in the number of events as self-important markers. The trained system consumes between 0.65s and 0.75s for detection and classification of 0.9s and 1s, respectively, emphasizing its use in real-time surveillance systems. In this regard, this methodology promises to provide helpful information for prompt decision-making based on micro-seismic monitoring.

Now, earthquake detection through ML techniques is not only limited to conventional seismic data/events but can also be done directly using scanned images (Wang *et al.*, 2019) [42], Distributed Acoustic Sensing (DAS) (Hernandez *et al.*, 2022) [43]. Low-frequency earthquake data (Thomas *et al.*, 2021) [44] and tremor (Hulbert *et al.*, 2020) [45]. Machine-learning earthquake detectors have come a long way, demonstrating that ML earthquake detectors can generate fewer false positives than STA/LTA and are more robust for low SNR signals. Nevertheless, computational demands have gone up, although it is now possible to do efficient real-time processing across different environmental variables.

In conclusion, the STA/LTA method is widely used to detect seismic waves, but ML successfully automated this task. Moreover, mixing data from different sources could help accomplish

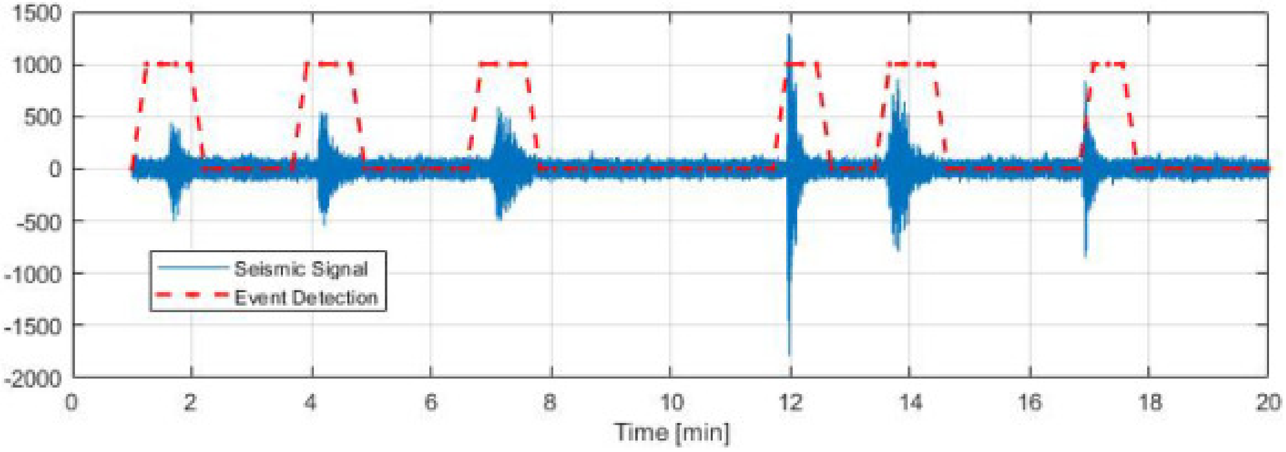


Figure 2.3: Seismic Signal Detection for a continuous time-history with six earthquake events. Source: Lara *et al.*, 2021 [41]

the event detection. Significant scientific endeavors in computer vision, notably the so-called *attention mechanism* and other recursive methods, outperformed traditional tools for seismic signal detection [46].

2.1.3 Phase picking

After detecting seismic events, the subsequent tasks involve determining, measuring, and identifying P-Wave and S-Wave arrival times. This task has been traditionally performed manually by expert seismologists:
It is a time-consuming task that requires an experienced analyst, and the amount of work can be overwhelming during swarms and aftershocks sequences when in formations flux is high., Mousavi *et al.*,2023 [18]
Temporal resolution is essential to perform an accurate phase picking. Historically, other than manual picking, classical approaches have relied on statistical and wavelet-based techniques. However, these methods are susceptible to noise and biases, and they encounter difficulties when identifying emerging S-Wave arrivals, as shown in Figure 2.4.

The introduction of AI to this field started in 2012 when Hammer *et al.*, 2012 [48] employed the HMM network architecture. This approach is suggested by Lara *et al.*, 2021 [41] for detecting seismic activity before a volcanic eruption. Instead, they used a few reference waveforms that are archived and learned from while recording such events just after an interesting event has been detected to create a classifier upon which a minimum number of them were based. Thus, one waveform can act as the key event classifier. Finally, this system and superficial neural network have also served as phase-picking tools.
Afterward, AI models have increased in complexity to improve accuracy with time. Different methods have been employed since then, including multi-station models made by Yand *et al.*, 2020 [49] and single-station models (Ross *et al.*, 2018), each with its own advantages and disadvantages. Mousavi *et al.*, 2023 [18] presented results that align with this observation. Notwithstanding, the generalizability of this model is uncertain, although it may help to reduce false favorable rates for a given symptom. Single-station models have benefited from extensive training data, leading to robust generalization. However, they are also vulnerable regarding hyperparameter tuning and the training process.
Nevertheless, an important thing about these methods is that due to their massive data set training, they can act as pre-trained models for seismic tasks without retraining or fine-

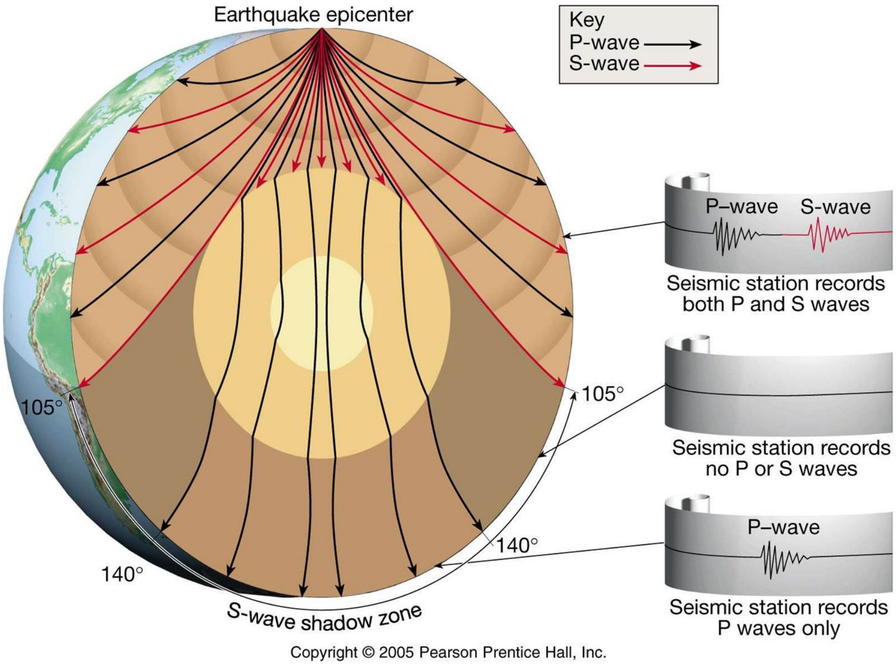


Figure 2.4: Arrival of P-Wave and S-Wave after an earthquake around the earth [47]

tuning [50]. Lapins *et al.*, 2021 [50] proposed techniques such as Transfer Learning and feature extraction methods that effectively utilized these machines optimally for phase-picking end goals (Woollam, 2022). Lastly, the Seisbench of Woollam *et al.*,2022 [51] was another milestone because it provided a toolbox containing several machine learning algorithms for phase picking (see Figure 2.5).

In 2023, Li Wei and colleagues [52] proposed Python's SAIPy packages. This is a solution for handling seismological data. Earthquake detection, magnitude estimation, seismic phase picking, and polarity identification are some tasks it covers. An essential part of SAIPy is the inclusion of a pre-trained neural network referred to as CREIME_RT, which is an upgraded version of CREIME. When given a 3-channel accelerogram as input, CREIME_RT has achieved successful accuracy above 99.8% in earthquake identification. It is an improvement that shows how this package can contribute to making seismic analysis capable of real-time seismic testing.

Hence, enhancing the accuracy of the detection and phase-picking stages within earthquake monitoring pipelines generates a more intricate and comprehensive earthquake catalog.

2.1.4 Polarity Determination

Polarity determination is used to study the focal mechanism and determine the location of seismic events. This can be done with diffraction stacking. The type of faulting during an earthquake is revealed by focal mechanism analysis. Travel time curves are used for Kirchhoff migration to pick the source location. This process allows researchers to compare this position with the one they get from the diffraction stack. It was a manual and laborious process, but it's now been automated thanks to machine learning. A variety of methods have been developed to apply machine learning to polarity determination. Rosse *et al.* 's supervised method (2018a) [53] and Hara *et al.* 's (2019) [54] both make use of convolutional layers that detect polarities that humans already identified in past researches. However, these two methods were overshadowed by Mousavi *et al.* 's unsupervised clustering approach (2019b) [55]. They demonstrated that models can classify polarities better without taking damage in quality and

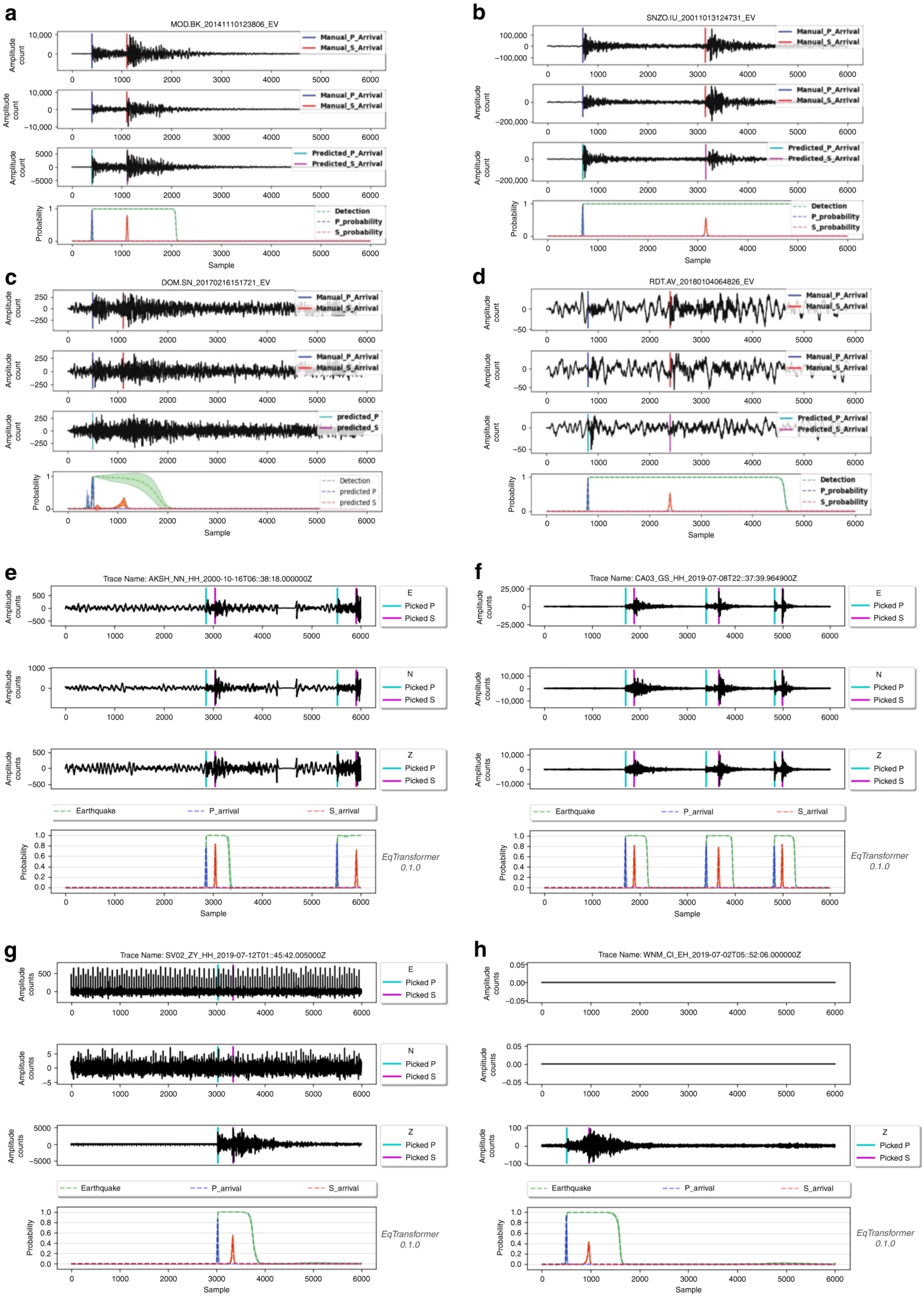


Figure 2.5: Illustration of detection of P-Wave and S-Wave using a neural network architecture, EQ-Transformer, [46]

achieve more accurate focal mechanisms, Uchide *et al.,* 2020 [56] (See Figure 2.6).

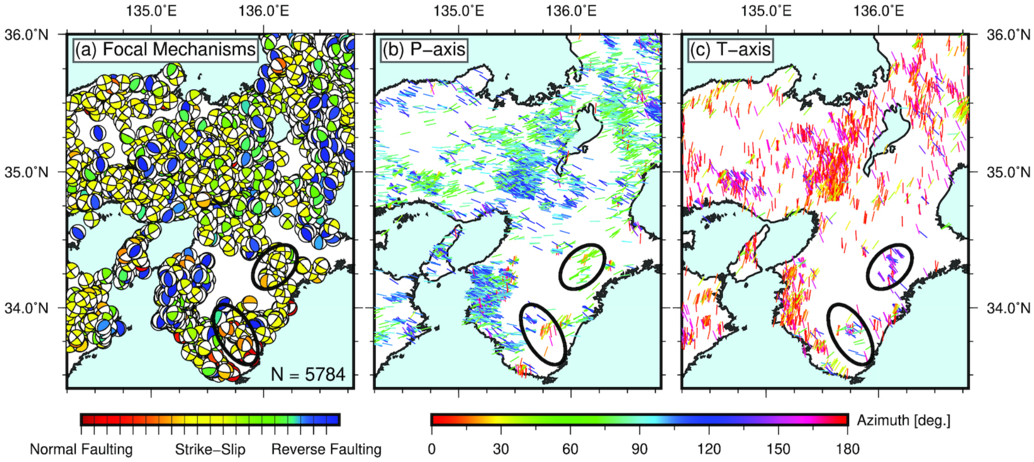


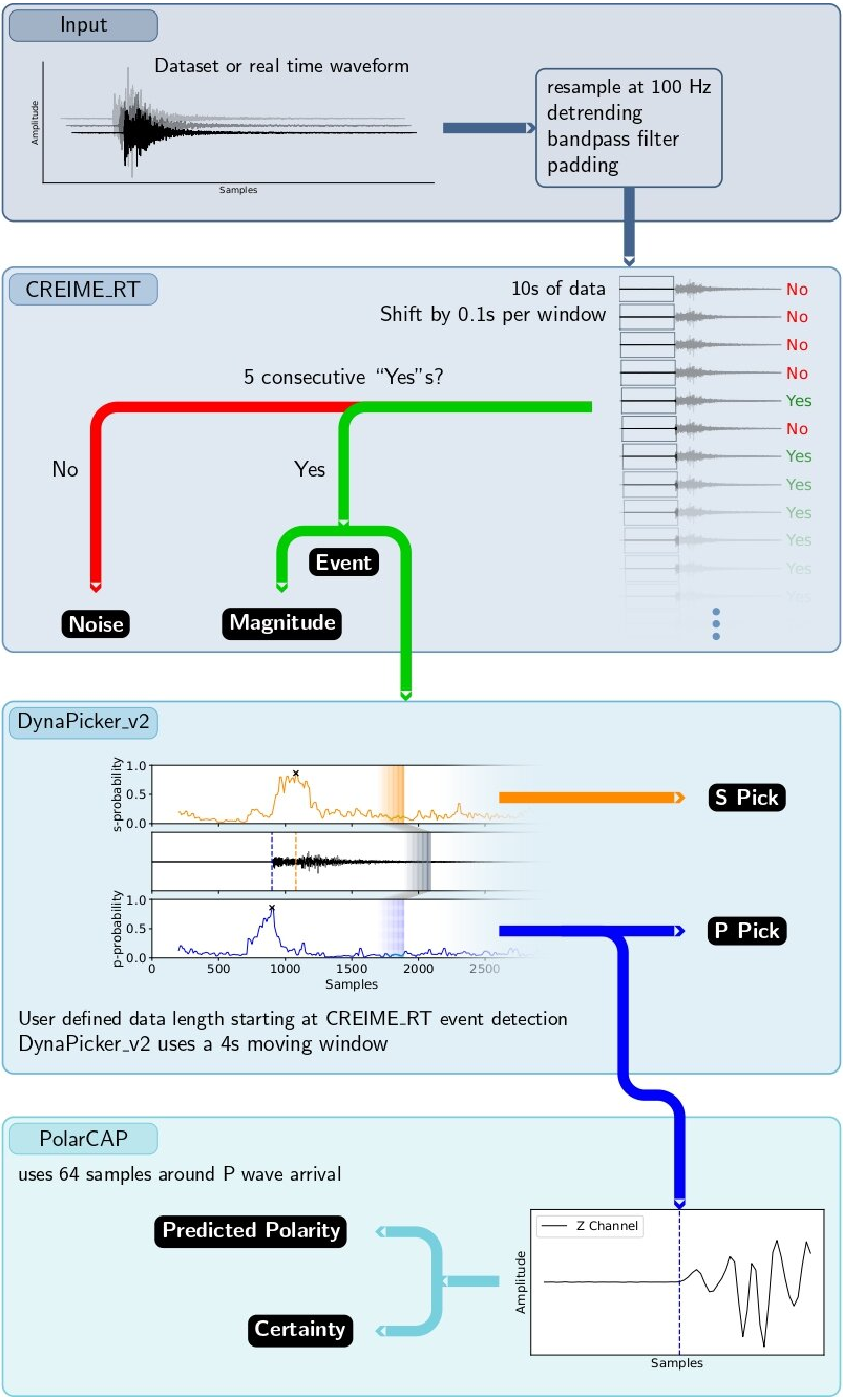
Figure 2.6: Focal mechanisms, (b) P-axes and (c) T-axes in the Kansai area. In (b) and (c), the axes with plunges less than 30° and the focal mechanism solutions ranked A–C were plotted. The ellipses indicate areas with distinct focal mechanisms. Source: Uchide *et al.*, 2020 [56]

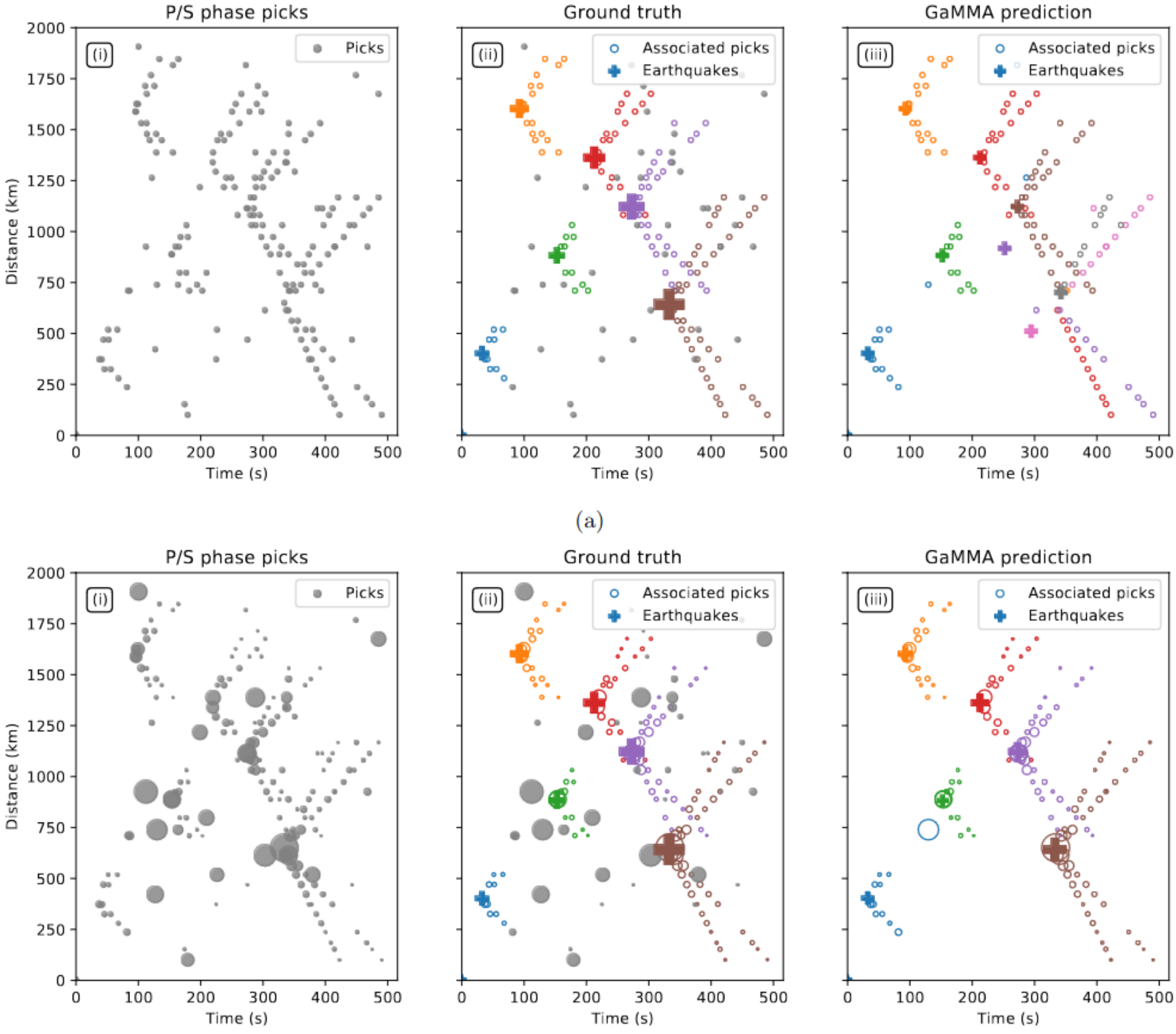
Uchide's previous work was only done in the context of Japanese seismicity. Now, using SAIPy [52], which was trained on the widened STEAD dataset, it has been possible to expand to more exhaustive and standardized events around the world. Including this new framework marks significant progress in generalization (see Figure 2.7). The 64 samples around P-wave arrival swift prepossessed were picked from CREIME_RT and used to predict polarity. This value then evaluated the uncertainty of the prediction. Therefore, the investigation made in Polarity determination underscores the efficacy of integrating machine learning. Even better, showing that incorporating this technology not only speeds up the process but also increases accuracy. This transition has proven how this handcrafted strategy could be done faster and more precisely using machine learning methodologies.

2.1.5 Phase Association

Phase association is a method used to link the seismic signal detected at different monitoring stations within a seismic network. This tool helps to detect the occurrence of an earthquake by identifying seismic phases at various locations and tracking their arrival times. In seismology, Phase Association is one of the crucial and challenging tasks, and the endeavors need more precision. Machine Learning approaches find a way of application. Time-based methods use the measurement of travel time for phase association. Two primary strategies are employed: pattern detection metrics and search-based associations. PhaseLink [57], which utilizes pattern detection metrics, has demonstrated the potential of employing a Recurrent Neural Network for phase association. This method simultaneously addresses source and phase association. In a similar vein, Zhu *et al.* (2022) [58] combined research on a grid and neural network techniques for phase association. The authors treated phase association as a probabilistic clustering problem within a framework where each earthquake corresponds to a cluster of P or S phases. These phases exhibit hyperbolic arrival time move-out and amplitude decay with distance, as shown in Figure 2.8.

Waveform-based methods explore waveform similarity from nearby earthquake sources within a low-dimensional feature space. For instance, Convolutional Neural Networks (CNNs) can be trained on pairs of earthquake waveforms recorded at two stations, specifically across ma-


Input
Dataset or real time waveform
Amplitude
Samples
resample at 100 Hz
detrending
bandpass filter
padding
CREIME_RT
10s of data
Shift by 0.1s per window
No
No
No
No
Yes
No
Yes
Yes
Yes
Yes
Yes
Yes
5 consecutive "Yes"s?
No
Yes
Event
Noise
Magnitude
DynaPicker_v2
s-probability
1.0
0.5
0.0
p-probability
1.0
0.5
0.0
0
500
1000
1500
2000
2500
Samples
S Pick
P Pick
User defined data length starting at CREIME_RT event detection
DynaPicker_v2 uses a 4s moving window
PolarCAP
uses 64 samples around P wave arrival
Predicted Polarity
Certainty
Z Channel
Amplitude
Samples



(a)

(b)

Figure 2.8: Synthetic example of phase association (courtesy of [58]): (a) association using only arrival times; (b) association using both time and amplitude. The left panels plot the P- and S-phase picks

jor phase arrival times, to predict whether they originate from a common seismic source. This research was conducted by McBreaty *et al.* [59]. Travel-time-based methods have been demonstrated to perform better phase association, but they are less generalizable, contrary to wave form-based methods [59].

2.1.6 Earthquake Source Parameters

The scientific community has labored to measure the parameters of earthquakes as they happen accurately. This is a huge problem when it comes to earthquake detection or prediction. Conventional methods are too slow and not accurate enough, so researchers turned to Machine Learning for help. More specifically, a method called PreSEIS (Pre-SEISmic [60], Bose *et al.*, 2008 [60]) was pioneered that could rapidly estimate seismic parameters shortly after an earthquake begins and before its end. It uses data from multiple sensors at different places to estimate where the earthquake's center is located, how big it is, and how wide the rupture will be. The method consists of two layers of feed-forward neural networks, which are much faster than traditional methods based on single stations. The work of Bose collaborators used PreSEIS on the Istanbul Earthquake Rapid Response and Early Warning System (IERREWS). Compared with other methods, there was no dispute, and PreSEIS outperformed them in terms of speed and calculation time [18]. Machines have been used in this area before, though with limited impact. Kaüf *et al.*, 2016 [61], have extended this work. They applied a nonlinear Bayesian method that relies on pattern recognition and synthetic 3-D Green's functions to quickly deduce point source parameters from strong-motion data [61]. By representing probabilistic inverse mapping using a deep neural network, they could provide probability distributions for source parameters, allowing scenarios to be efficiently mapped with minimal CPU and memory resource demands. More recent approaches are described in the following paragraphs.

(i) **Single-station methods** : The real-time seismogram analysis has traditionally relied on well-established, rule-based seismological procedures to elucidate the characteristics of seismic events, including earthquakes and tsunamis. However, early characterization of such events is often not directly possible, and the performance of machine learning has been used to overcome that situation. For the feasibility of that technology, Lomax *et al.*, 2019 [62] have focused their attention on identifying seismic parameters from data provided by a short-station waveform. Their study demonstrates that convolutional neural networks (CNNs) can directly detect information with minimal pre-processing, eliminating the need for extensive feature extraction. The continuity of this work can be found in the publication of Mousavi *et* Beroza., 2020a [63]. They used a multi-task temporal-convolutional neural network to learn epicentral distance and P travel time from 1-minute seismograms. The network provided estimates for epicentral distance and P travel time with errors of 0.23 km and 0.03 s, along with information about their uncertainties. The previous development paved the way for further exploration. Ristea *et al.,* 2021 [64] improved the work of Mousavi in the case of single-station-based estimation of epicentral distance, depth, and magnitudes by using a time-frequency representation of 3-channel waveform as input and utilizing CNN (see Figure 2.9).

(ii) **Multi-station methods**: The approach takes data from numerous stations in a seismic network and extracts patterns in how seismic energy propagates across it. Plenty of researchers have already delved into this method, known as a multi-station method. Most recently, Licciardi *et al.*, 2022, for EEW (Earthquake Early Warning), created a deep neural network that analyzed prompt elasto-gravity signals (PEGS) to predict both an

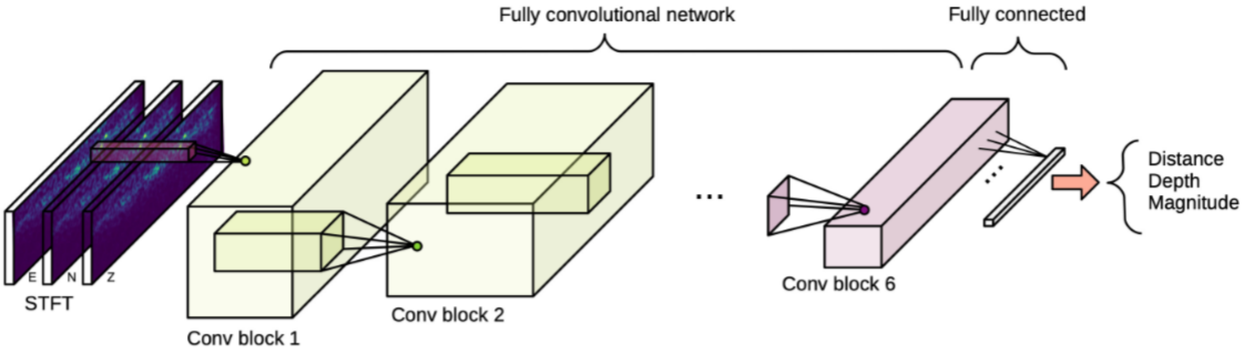


Figure 2.9: This is the complete architecture of a CNN model. The input data are a 3-channel tensor of three concatenated STFTs (one transform on each direction EW, NS, and UD). The input is processed through a series of Convolutional layers. At the end of the process, the distance, depth, and magnitude of the signals are extracted. Source: Ristea *et al.*, 2021 [64]

earthquake's location and its magnitude ([65]). The team trained their model on synthetic data, and using only PEGS recorded during the 2011 Tōhoku earthquake, it showed that such networks could estimate the magnitude with a final M_w of 9 ± 0.05, using only 315 s of data. *"After around 55 s, PEGSNet is always closer to the 'true' STF than the other algorithms"*,[65]. As a result, PEGS has surpassed the speed of the current state-of-art EEW system.

Kriegerowski *et al.*, 2018 [66] introduced a novel method for earthquake location determination, called *multistation regression.* Traditionally, neural networks have been trained on single-station records, showing some limits. To overcome these limitations, the present approach used 3-component waveforms from multiple stations to avoid losing information during the pre-processing. Collected data was adopted with no preliminary preparation besides record alignment based on the P-wave arrival time at the reference station. Therefore, Kriegerowski and collaborators were able to train their deep neural network and extract the pattern that determine the earthquake hypocenter locations.

On the other hand, Zhang *et al.*, 2020 [67] considered earthquake detection by framing this latter as an image classification problem. They recognized that convolutional neural networks, widely used in image processing tasks, can effectively address this challenge. Therefore, they trained a CNN model jointly using records of multiple stations. Their team introduced a similar multi-station approach but with a twist. They formulated it within a classification framework, where they represented a 3D grid covering a specific region of interest using 3D kernels at the output layer of a CNN. The nearest grid point to the earthquake hypocenter was labeled in this setup using a 3D Gaussian function.

Lin *et al.* (2021) [68] used this approach to solve the problems posed by EEW systems; due to both the rarity the complexity of significant earthquake events, it becomes challenging to characterize the magnitude, depth and different parameter of a seismic event. The developed model, denoted as M-LARGE (shown in Figure 2.10), is specifically designed to address this problem. More than that, in the classical systems, determining the magnitude of significant earthquakes was underestimated. In their study, the proposed M-LARGE examines and determines the relevant patterns of crustal deformation in real-time, enhancing EEW. Trained over 6 million different simulated rupture scenarios recorded on the Chilean GNSS network, this method attains a rate of accuracy of 99%.

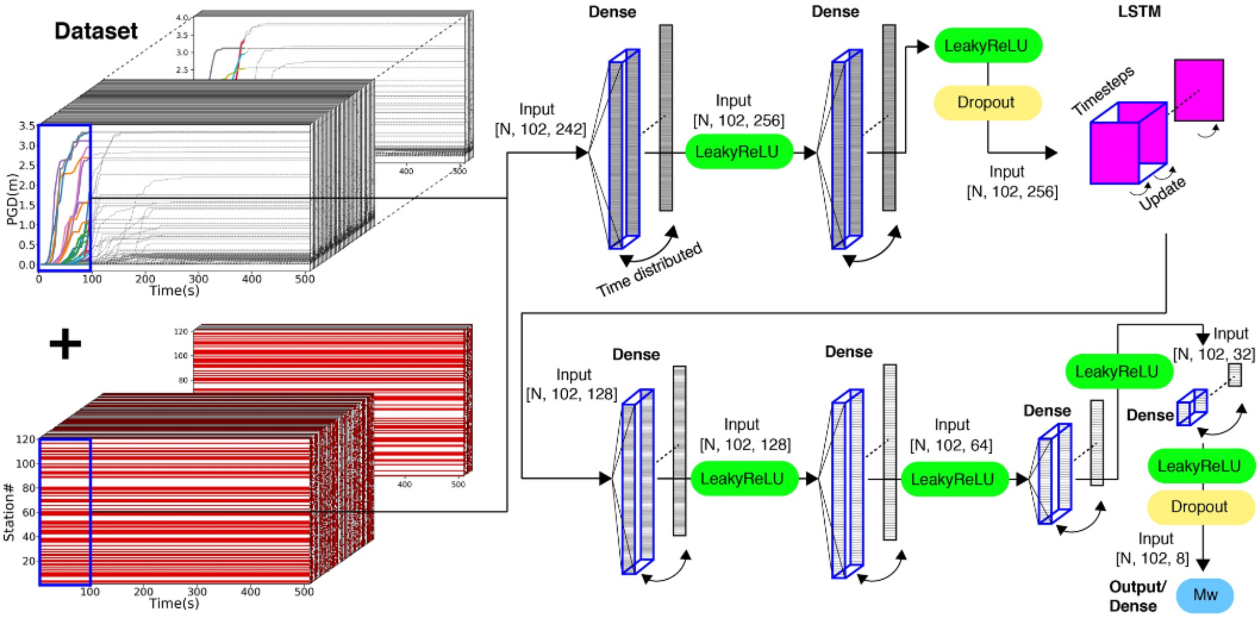


Figure 2.10: M-LARGE model architecture, showing the input as the time-dependent PGD (Peak Ground Displacement) values from GNSS stations. Courtesy of Li *et al.*, 2021 [68]

"Furthermore, the model successfully predicts the magnitude of five real Chilean earthquakes in the last 11 years" [68]. To conclude, the described research has proved that the multi-station method is a promising method for seismology in the future. Nevertheless, in multi-station-based methods, large input and output dimensions increase the network architecture's complexity and nonlinearity, leading to a difficult and onerous training process with a limited number of multi-labeled stations. This limitation is treated by *dynamic multi-station methods.*

(iii) **Dynamic multi-station methods**: To tackle the limitation of the previous studies, the works of Van Den Ende and Ampuero *et al.*, 2020 [69] proposed a multi-station approach based on edge-less graph neural network. This graph neural network (GNN) approach explicitly incorporates spatial information. The location, underestimated mainly in the previous machine learning, is used by Van Den Ende *et al.* to improve the performance of their machine learning model. The GNN is designed for seismic source characterization (specifically, location and magnitude estimation) based on multi-station waveform recordings. The results showed that the proposed approach achieves superior model prediction accuracy compared to other location-agnostic methods. It has been proven to be highly adaptable to varying numbers and arrangements of seismic stations, offering new possibilities for automation in seismological tasks and improving earthquake early warning systems.

Münchmeyer *et al.*, 2021a [70] introduced an innovative model for real-time earthquake detection and location estimation. They employed a specialized neural network architecture called the Transformer (more detail in Section§ 2.4.5). This model effectively incorporates waveform data from a dynamically changing array of stations. Their empirical investigation conclusively demonstrated that this novel approach surpasses conventional methods in parameter estimation.

Chin *et al.*, 2021 [71] proposes an approach which, instead of using raw waveforms, relies on phase sampling times and triggered station positions as input data. They use an attention-based hypocentre estimation model known as AHE. The attention mechanisms have the ability of long-range correlations in the input sequences.

These studies conclusively demonstrated that classical CNN-based neural networks should be better suited to this task due to the need to consider long-range dependencies in wave propagation. We also raised the sequential nature of seismic waves. These data need to be considered in terms of time and space. This has been advantageously taken into account by transformer-based networks.

(iv) **Physic-informed methods** In 2022, Smith *et al.* [72] obtained interesting results in seismology by adopting a different approach that represents a breakthrough in the field. Instead of using deep neural networks (DNNs) for feature extraction or employing attention techniques to capture long-range dependencies, they proposed using a distinct type of neural network known as a Physics-Informed Neural Network (PINN). PINN, developed by Maziar Raissi [73] and his collaborators, first proposed a way to incorporate the laws of physics into the traditional neural network framework. Thanks to PINNs, neural networks were first employed to solve partial differential equations (PDE). The core idea behind it is to minimize the Mean Square Error (MSE) of discretized PDE with classical discretization (based on Taylor's limited expansion):

$$\begin{aligned} \left.\frac{d^2u}{dx^2}\right|_{x_i} &= \frac{u\left(x_i+\Delta x\right)-2u\left(x_i\right)+u\left(x_i-\Delta x\right)}{(\Delta x)^2}+O\left((\Delta x)^2\right) \\ \left.\frac{du}{dx}\right|_{x_i} &= \frac{u(x_i)-u\left(x_i-\Delta x\right)}{(\Delta x)}+o\left(\Delta x\right) \end{aligned} \tag{2.1}$$

The prediction will be more robust if the physics law is followed.

For example, if we train a PINN for a harmonic oscillator whose EDP has the expression:

$$m\frac{d^2u}{dx^2}+\mu\frac{du}{dx}+ku=0 \tag{2.2}$$

The cost function will be

$$\min\frac{1}{N}\sum_{i}^{N}(u_{\mathrm{NN}}(x_i;\theta)-u_{\mathrm{true}}(x_i))+\frac{1}{M}\sum_{j}^{M}\left(\left[m\frac{d^2u}{dx^2}+\mu\frac{du}{dx}+ku\right]u_{\mathrm{NN}}(x_i;\theta)\right)^2 \tag{2.3}$$

PINN succeeded in several fluid and solid mechanics applications to solve partial derivative equations without numerical simulations. In Seismology, [74] proposed an architecture – denoted HypoSVI – that employs PINN framework to solve the travel-time Eikonal equation that allows finding the real source position x, realizing the maximum log-likelihood function of the source location, namely, $\ln p(x)$, as shown in panel 1 in Figure 2.11. A set of particle locations x_i (black dots in Figure 2.11) is drawn at training epoch 1 and updated at each epoch by minimizing kernelized Stein discrepancy between the parametrized likelihood and the observed source position distribution. Once the initial particle locations are supplied, the predicted travel times are determined from the distance between all particle locations to all observation points (red triangles in Figure 2.11). Using the solutions of these differential equations, they deduced the hypocentres from the observed arrival times. This innovative strategy demonstrated robustness in handling the common occurrence of high multi-modal posterior distributions in hypo-central inverse problems.

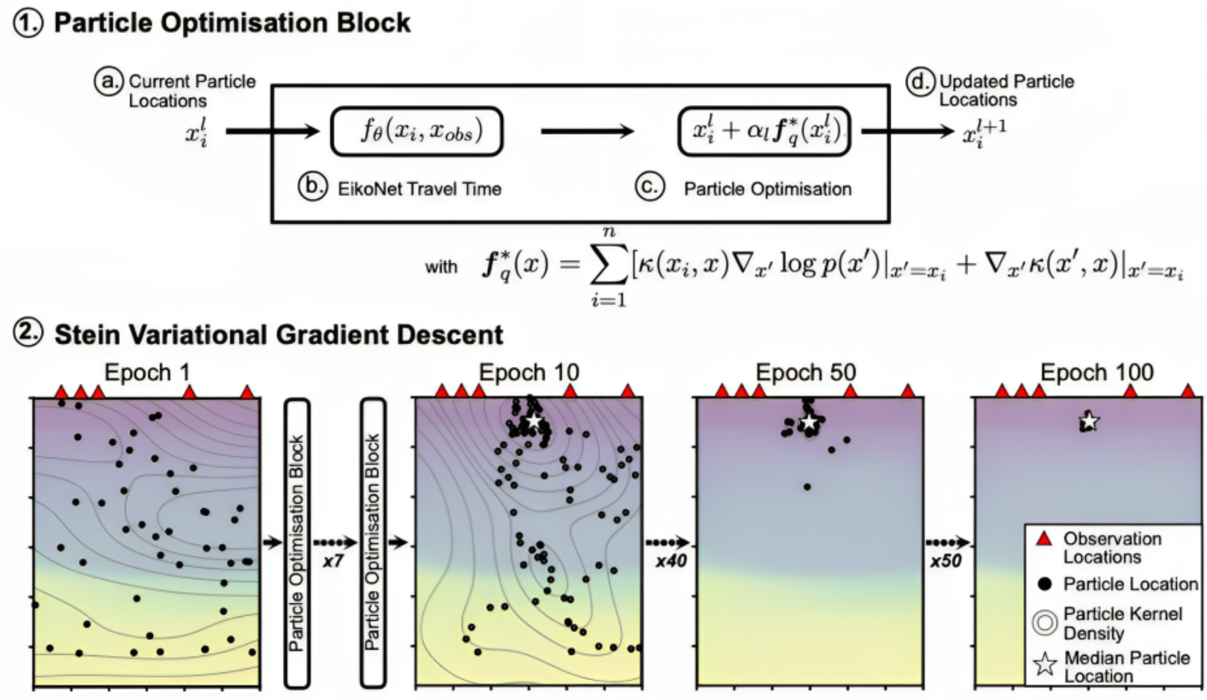


Figure 2.11: Panel 1: HypoSVI training scheme. Panel 2: example of progressive Stein variational gradient descent. Black dots: particle positions. Red triangles: observation points. Contours represent the particle kernel density, whereas the white star represents the median location of the particles, representing the optimal hypocentral location. Source: Smith *et al.*, 2022 [72]

(v) **Earthquake source mechanism** Another crucial task in seismology is to quickly provide an automated report on the mechanism at the source focus after a seismic event, particularly a destructive one. As a practical matter, this should enable rapid characterization of fault geometry, assessment of stress perturbations, and underlying aftershock patterns. To this end, Kuang et al., 2021 [75] proposed Focal Mechanism Network (FMNet), a new model to attempt to solve this problem. This method quickly evaluates and estimates focal mechanisms based on multi-station dynamics. Trained with 787320 synthetic samples, FMNet estimated the focal mechanisms of four Ridgecrest earthquakes in 2019 of magnitude greater than M_W 5.4.

Steinberge *et al.*, 2021 [76] use a Bayesian Neural Networks (BNN) to *”rapidly estimate full moment tensors of earthquakes and their parameter uncertainties“*. This is a probabilistic approach to detect focal mechanisms. This approach trains individual Bayesian neural networks for each grid point in a 3D regional mesh using synthetic seismic waveforms and moment tensor parameters. They successfully applied the BNN as a proof of concept on seismic waveform recordings of aftershocks of the Ridgecrest 2019 earthquake with moment magnitudes ranging from M_W 2.7 to M_W5.5.

Zhang et al. 2021 [77] introduced an innovative physics-guided neural network tailored for estimating shear-tensile focal mechanisms. This technology was designed explicitly for micro-earthquakes, utilizing displacement amplitudes of direct P waves recorded on shallow-borehole arrays.

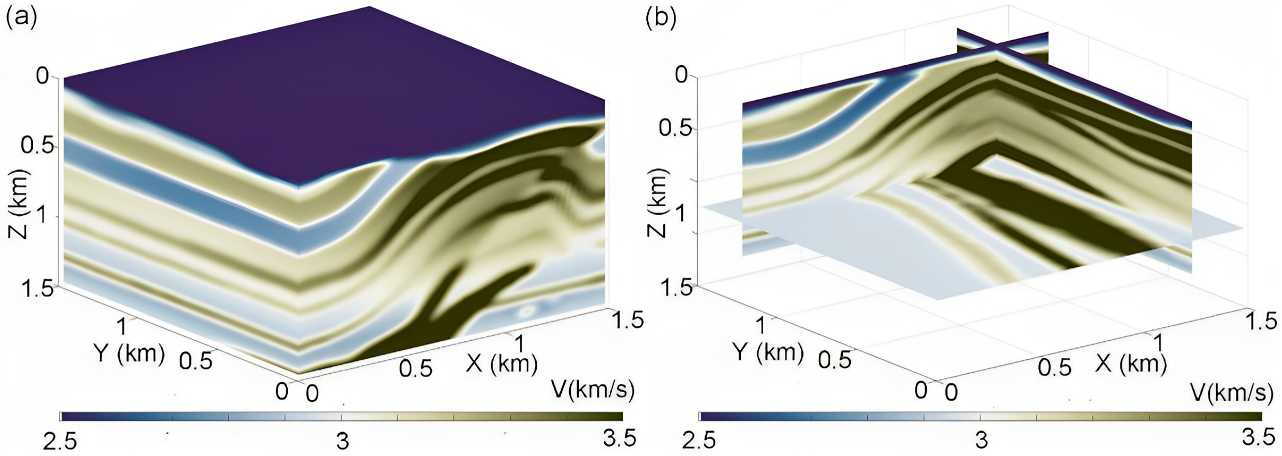


Figure 2.12: PINN model proposed by Song *et al.*, 2020 [79]. (a) The geological velocity model; (b) three corresponding slices of the modified overthrust model obtained by PINN.

2.1.7 Synthetic Seismogram Simulation

Several methods, such as FEM and SEM, have been used for some time to simulate wave propagation. FEM and SEM have proven effective for simulation wave propagation in heterogeneous media. Such techniques provide an accurate approximation of the differential equation of the problem. However, despite their mathematical rigor, they face some limitations. The multiscale nature of certain problems poses a numerical challenge. For large-scale model simulations, difficulties quickly arise as computational costs increase steadily. In therm of wave propagation, frequencies above 5 Hz seem unreachable. This is where machine learning comes in. There are two main approaches: one based on Physical Information Neural Network and the other on Generative Adversarial Network (GAN).

PINN was designed to solve the partial differential equation by minimizing the residual between the ground truth at the monitor point and the inferred solution (via neural network) at the exact location. Space and time derivatives are easily computed, adopting the automatic differentiation libraries developed for stochastic gradient descent in machine learning. In this context, Mosely *et al.*, 2020 [78], successfully introduced PINN as a strategy to learn the solution of the wave equation and introducing the boundary conditions as a penalty in the loss function on the residual between synthetic and observations. Testing 2D acoustic wave equations with varied velocity models, from simple to realistic, demonstrates the network's accuracy. The physics constraint in the loss function enables the network to solve the wave equation accurately close to the boundaries. Song *et al.*, 2020 [79] proposed a similar strategy in the frequency domain. As shown in Figure 2.12 the authors utilized a modified wave equation for transversely isotropic media with vertical symmetry as the loss function to train the PINN. They focused on solving for the scattered pressure wave field instead of the overall wave field. Applying this model, they derived frequency-domain acoustic solutions in 2D and 3D scenarios, considering irregular topography and maintaining a fixed source location. The numerical experiments proved that PINNs are relatively efficient and accurate, drastically reducing the computational time needed to obtain the solution with numerical methods.

Generative Adversarial Network (GAN), GoodFellow *et al.*, 2014 [80], is designed to produce indefinitely fake samples that closely resemble the original ones. This task leverages the approximation power of neural networks to learn the probability distribution of attributes in the trained data by optimizing a generator through a discriminator. Further exploration of this concept is available in Section 2.3.3. GAN has proven its utility in various applications, such as

Nomenclature	**Description**	**Unit**	**Formulation**
PGA	Peack Ground Acceleration	g	$PGA = \max_t \lvert \ddot{u}_g(t) \rvert$
PGV	Peak Ground Velocity	cm/s	$PGV = \max_t \lvert \dot{u}_g(t) \rvert$
S_a	Spectral acceleration	g	$S_a(T_i) = \omega_i^2 S_d(T_i)$
I_A	Aria Intensity	cm/s	$I_A = \frac{\pi}{2g} \int_0^{T_d} \left[\ddot{u}_g(t)\right]^2 dt$
I_V	Velocity Intensity	cm	$I_V = \frac{1}{PGV} \int_0^{T_d} \left[\dot{u}_g(t)\right]^2 dt$
CAV	Cumulative absolute velocity	cm/s	$CAV = \int_0^{T_d} \lvert \ddot{u}_g(t) \rvert dt$
CAD	Cumulative absolute displacement	cm	$CAV = \int_0^{T_d} \lvert \dot{u}_g(t) \rvert dt$
ASI	Acceleration spectrum intensity	cm/s	$ASI = \int_{T_i}^{T_d} \mathrm{SA}(T, \xi) dT$

Table 2.1: Intensity definition. $g = 9.8 m/s^2$ represent the gravity acceleration

seismograms data augmentation, see Wang *et al.*, 2020 [81], Wu *et al.*, 2021 [82]. It can generate broadband seismic signals by combining the low-frequency results from physics-based numerical simulations with sparsely sampled broadband observations (Gatti *et al.*, 2020 [21]). Additionally, GANs can generate three-components strong motion time-series for different magnitudes, distances, and $V_{S,30}$ values (Florez *et al.*, 2022 [83]). They are also used to refine computer simulations of Distributed Acoustic Sensing (DAS) data for ground motion noise, Hermandez *et al.*, 2022 [43]. In a nutshell, GANs offer an efficient framework for generating large-scale synthetic training data, enhancing the performance of deep-learning classifiers, see Li *et al.*, 2020 [84].

In this sense, ANN2BB, developed by Paolucci and his collaborators, 2018 [85] has proven the possibility of finding a strong "correlation between short and long period spectral ordinates trained on strong motion record", going beyond the limits of the classical deterministic numerical simulation. ANN2BB allows the enhance the outcome of synthetic numerical simulations (valid for periods above 0.75 s) to generate hybrid broad-band synthetic (valid for periods above 0.05 Hz) by inferring the short-period pseudo-spectral acceleration (PSA) ordinates (0.05- 0.75 s) from the long-period part ($T >$0.75 s) numerically simulated. Hybrid PSA spectra (numerical simulations for the long-period part, ANN for the short-period part) are adopted as targets to iteratively scale the numerically-simulated long-period time histories adding random-phase Fourier amplitudes following the Sabetta and Pugliese method [86]. ANN2BB was successfully adopted by Gatti *et al.* [16] to render hybrid strong ground motion scenarios of the M_W6.7 2007 Niigata-Chuetsu-Oki earthquake in Japan. The free-field hybrid ground motion generated by ANN2BB was based on numerical simulations of the Niigata earthquake on a 50 km × 68 km vast region, including the nuclear site of Kashiwazaki-Kariwa. The latter was chosen as a case study for an international benchmark. Gatti *et al.* [16] employed the free-field hybrid ground motion to perform a subsequent Soil-Structure Interaction study, approximating the structural response of one reactor building during one major aftershock following the Niigata 2007 main shock.

2.1.8 Ground Motion Characterization

Ground motion is usually characterized by its Intensity Measures (IM), such as Peak Ground Acceleration (PGA), Peak Ground Velocity (PGV), Cumulative absolute velocity (CAV), pseudo-spectral acceleration (PSA), and Arias Intensity. These parameters can be determined using ML. For instance, Machine Learning demonstrated its effectiveness in constructing non-parametric Ground Motion Models (GMM), mainly through (deep) neural networks. In this context, the neural network represents the model coefficient, and a fixed set of similar input parameters is utilized, see Table 2.1.

In the work of Derras *et al.,* 2014 [87], a neural network was used to develop a ground motion prediction equation (GMPE) for Europe in a model with only five inputs-magnitude, Joyner-Boore distance, focal mechanism[1], hypocentral depth and $V_{S,30}$ [2] and outputs of PGA, PGV, and 5% damping at period from 0.01 to 4 s.

Hu *et al.* (2022) [89] adopted Support Vector Regression (SVR), focusing on minimizing the generalized error bound for enhanced robustness, to formulate Ground Motion Models (GMMs) for Arias intensity, CAV, and significant duration. Their primary objective was to perform a regional seismic hazard analysis for the earthquake-prone Kanto region in Japan. To evaluate the rationality and effectiveness of the SVR GMMs, Hu and collaborators have analyzed performance indices such as correlation coefficients, slope coefficients, and residuals. Remarkably, the residuals of the SVR GMMs displayed no significant deviation concerning magnitude, rupture distance, or shear-wave velocity. Moreover, the standard deviations of model residuals, calculated from the regional ground motion database, were observed to be lower than those derived from previous models based on either Japanese or global databases. The study additionally compared the SVR GMMs with observed data and previous GMMs. The data-driven SVR method is commended for its capacity to develop GMMs without constraints from specific mathematical forms, addressing concerns about the potential impact of the structure of previous models on prediction performance. This approach effectively captures regional attenuation characteristics in seismic activity within the Kanto region.

Withers *et al.* (2020) [90] conceived a model to build GMPE from a database of ground motion simulations. Their findings indicate that the neural network–based GMPE provides bias and variability similar to empirical GMPEs for the Ridgecrest event. This suggests that machine learning-based GMPEs have the potential to emerge as potent tools in the next generation of seismic hazard models.

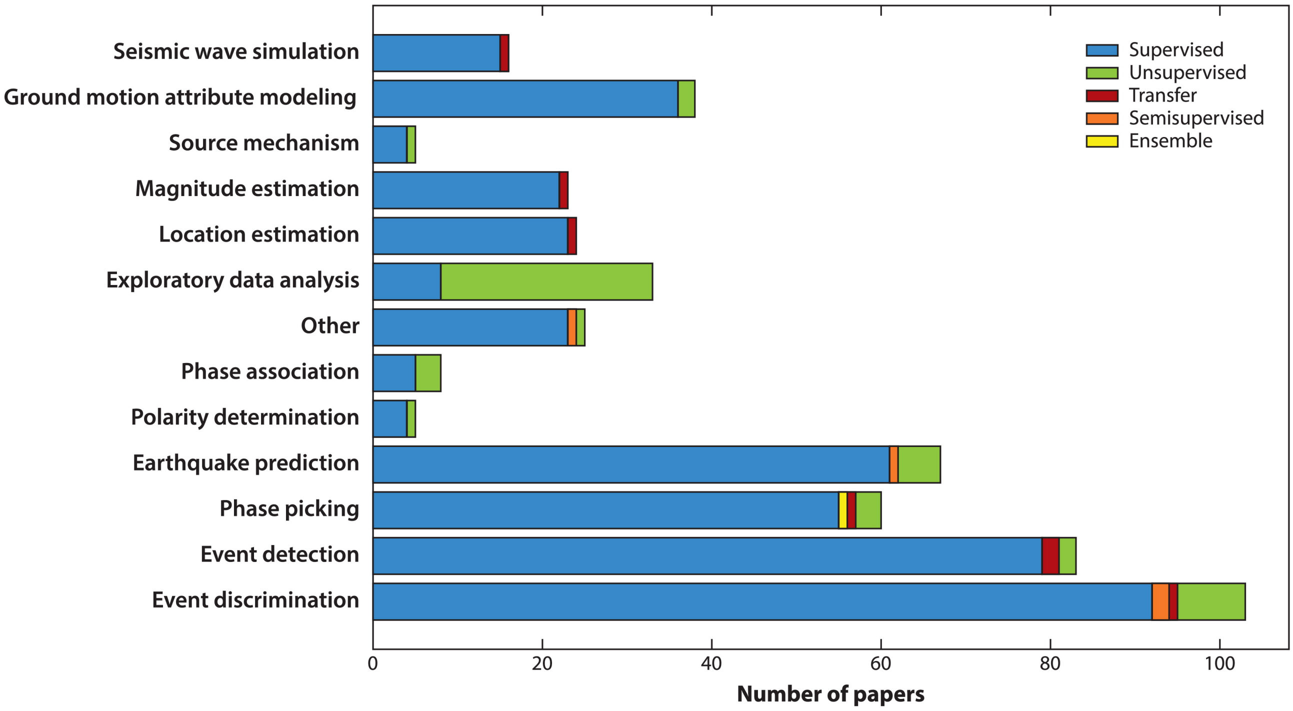
Other authors have approached this task using more modern architectures. Hsu & Huang, 2021 [91] employed Convolutional Neural Networks (CNN) to predict peak ground motion at a station, relying on the first 3 seconds of waveform data observed at the station. Their work extends beyond mere prediction, demonstrating the capability to forecast PGA for short and straightforward earthquakes.

Münchmeyer *et al.*, 2021b [70] developed an end-to-end approach for estimating PGA (Peak Ground Acceleration) probability densities at specific locations. This estimation relies on observed waveforms from a designated set of seismic stations. Their methodology showcases flexibility, allowing for an arbitrary number of stations at any location by implementing station-level feature extraction and PGA prediction.

---

[1] Seismologists define the focal mechanism of an earthquake as the direction of slip on the associated fault. They use symbols such as the "beach balls" on maps, representing the orientation of the stress field at the moment of rupture. The calculated focal mechanisms show the stress and pressure axes without the use of shading (`https://web.archive.org/web/20190601164109/https://earthquake.usgs.gov/learn/topics/beachball.php`).

[2] Defined as the harmonic mean wave velocity average over the 30 m directly below the ground [88]



Mousavi SM, Beroza GC. 2023
Annu. Rev. Earth Planet. Sci. 51:105–29

Figure 2.13: Summary of various approaches using machine learning for the different tasks and modeling in seismology. Source [18]

Partial conclusions on ML in seismology

The introduction of machine learning in seismology this past decade has changed how seismologists and earthquake engineers perceive these fields. The availability of large databases and the power of the machine learning model to absorb data and find relationships inherent to this have played a pivotal role. Subsequently, significant advancement has emerged and coincided with breakthrough technologies, such as Convolutional layers, Transformers networks, and improved graphic cards of higher speed and capacities.
Also, the information retrieved and collected from sensors and the ingeniously designed models have contributed to earthquake detection and early earthquake warning (EEW) systems. Phase picking, polarity determination, and source parametrization have benefited mainly from those approaches, and as a byproduct, synthetic databases have enriched and completed the recorded databases. Furthermore, standard numerical methods such as Finite Element Method (FEM) and Spectral Element Method (SEM) found their equivalence in machine learning through the Physics Informed Neural Networks (PINN) designed to solve wave equation and *Eikonal* equations. Finally, the recent application of Generative Adversarial Networks addresses the challenge of poor-quality signal reconstruction by enhanced physics-based simulation (PBS). All those contributions have then proved how artificial intelligence is significant for the future of seismology. Figure 2.13 summarizes the main machine-learning-related scientific activities in seismology and earthquake engineering.

2.2 Dimensionality reduction for multivariate data

When a dataset exhibits numerous features, many variables, and many realizations, interpreting which components are relevant for analysis can become challenging. Dimensionality reduction, therefore, emerges as reducing the number of dimensions or features while preserving essential information. This approach is helpful in statistic, particularly in machine learning, for several reasons:

1. The dimensionality reduction accelerates algorithm development and improving grouping and clustering [92];
2. These methods enhance classification accuracy [93];
3. Also, these methods enable effective visualization [94];
4. And these methods preserve regression information [95]

2.2.1 Principal Component Analysis

The foundation framework of Principal Component Analysis (PCA) traces its origins back to the seminal work put forth by Karl Pearson in 1901, as documented in his publication [96]. This method was initially applied to physical, statistical, and biological research. It is pertinent to note that a contemporaneous development, namely Exploratory Factor Analysis (EFA), was conceived by Charles Spearman in 1904, specifically to address psychometric assessments. During its early stage, the terms PCA and EFA were employed interchangeably within the domain of measurement theory. However, in subsequent years, the PCA methodology underwent a process of distinctive popularization, emerging as an elegant, straightforward technique to encapsulate and succinctly delineate complex data sets.

Principal Component Analysis (PCA) is employed on complex, multi-component data sets that inherently elude direct observation in low-dimensional spaces such as 2D or 3D. PCA outlines both auto- and cross-correlations within the data set. This procedure facilitates the representation of each entity within the sample space as discrete points within either a 2D or 3D geometric framework [97, 98]. All these points coexist within a unit-radius sphere. Consequently, this technique empowers identifying and isolating potential outliers, which may stem from sampling aberrations. As a result, it becomes feasible to discern clusters of samples corresponding to distinct classes when projected onto the principal component planes, as shown in Figure 2.14.

Principal Component Analysis (PCA) serves as a mechanism for dimensionality reduction, enabled by aggregating contributions from the primary dimensions. This methodology permits prioritizing dimensions that encapsulate 95% of the data set information. However, in the context of this thesis, PCA was not directly applied to seismic signals but rather to their encoded representations, *i.e.*, the latent space. This choice is motivated by PCA's significant computational demands, increasing with the dimension of the data. In the case of this thesis, owing to the utilization of approximately 30 7200 time-series, each comprising 4096 discrete time steps, the use of PCA was prohibitive. However, PCA will be preemptively leveraged to ascertain the optimal dimensions for the latent space. This strategic approach provides a precise gauge for the judicious selection of latent space vector dimensions. Moreover, PCA has been harnessed to evaluate the extent of potential intermingling among sub-classes, thereby affording the means to govern signal transformations. Further elaboration on this matter will be provided in the ensuing chapter.

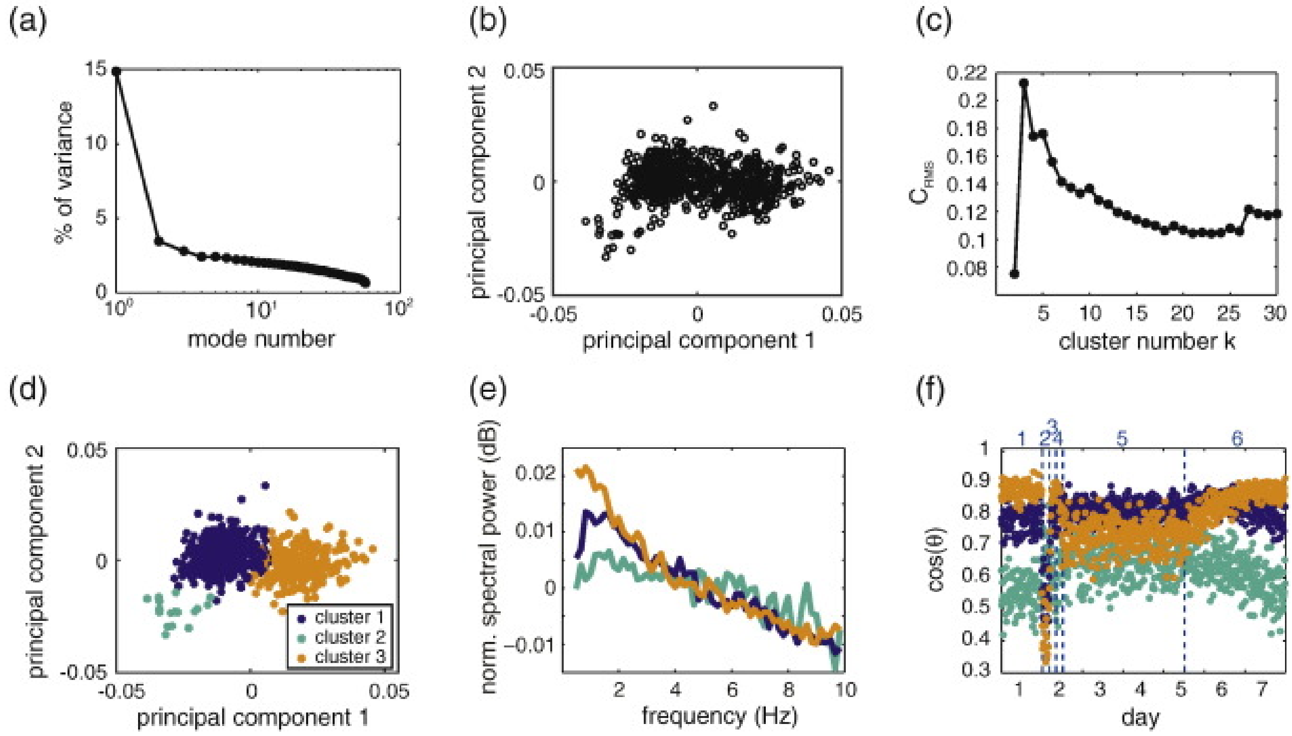


Figure 2.14: Unglert *et al.*, 2016 [99] in their study use PCA for pattern recognition in volcano seismic spectra."*PCA results for noise level 1.5. (a) Percentage of variance explained by each mode. (b) Projection of the observations in space spanned by principal components 1 and 2. (c) Evaluation measure for different cluster configurations, with the peak at k = 3 indicating the best cluster number. (d)–(f)*"

2.2.2 t-SNE and UMAP

Despite the undeniable advantages of PCA, it is imperative to recognize its inherent limitations when disentangling latent structures in high-dimensional data spaces. Depending on the sampling density, some data points or subgroups may have reduced 'observability,' leading to higher principal components associated with high uncertainty. In response to this challenge, techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) have been introduced [100]. These methods were designed to circumvent the aforementioned limitation associated with PCA by capitalizing on manifold learning principles.

On the one hand, the t-SNE was introduced by Geoffrey Hinton and Laurens van der Maaten in 2008 [101]. "*The technique is a variation of Stochastic Neighbor Embedding* [...] *that is much easier to optimize and produces better visualizations by reducing the tendency to crowd points together in the center of the map*". This tailored algorithm is designed to more effectively separate points that a straight line cannot separate. The formulation to evaluate the distance between two distributions p_{ij}, q_{ij}. Each distribution evaluates the distances with two points. In this theory, the distribution is assumed to be Gaussian with a standard deviation of σ_i. The high dimensional data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, in the targeted two or three dimension data

$\mathcal{Y} = \{y_1, y_2, ..., y_n\}$ that can be display with scatter plot.

$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq l}\exp\left(-\|x_k - x_l\|^2/2\sigma_i^2\right)}$$
$$q_{ij} = \frac{\left(1+\|y_i - y_j\|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\|y_k - y_l\|^2\right)^{-1}} \tag{2.4}$$

To optimize t-SNE, use Kullback-Leibler divergence

$$C = D_{\mathrm{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right) \tag{2.5}$$

is derivative is computed according to the following expression:

$$\frac{\partial C}{\partial y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|2)^{-1} \tag{2.6}$$

as shown in the Algorithm 1. On the other hand, UMAP was introduced by Leland McInnes and John Healy in 2018 [102] to circumvent the lack of precision. As shown in Figure 2.15, UMAP and t-SNE are comparable for visualization quality. However, UMAP better preserves the global data structure and has a cheaper computational cost than t-SNE. Furthermore, UMAP has no computational restrictions on the embedding dimension, making it viable as a general-purpose dimension reduction technique for machine learning, as shown in the example Figure 2.16. The formulation of the UMAP cost function :

Algorithm 1 t-SNE algorithm. Source [101]

```
**Data** : data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$
cost function parameters: perplexity Perp
optimization parameters: number of interations, $T$, learning rate, $\eta$, momentum, $\alpha(t)$
**Result**: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$
Compute pair wise affinities $p_{j|i}$ with perplexity : $Perp(P_i) = 2^{H(P_i)}$, with $H(P_i) = -\sum_j p_{j|i}\log_2 p_{j|i}$
Set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
sample intial solution $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
**for** $t = 1$ to T **do**
    compute low-dimensional affinities $q_{ij}$
    compute gradient $\frac{\partial C}{\partial \mathcal{Y}}$
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta\frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$
**end for**
```

$$C_{UMAP} = \sum_{i\neq j} v_{ij}\log\left(\frac{v_{ij}}{w_{ij}}\right) + (1 - v_{ij})\log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right) \tag{2.7}$$

In this formula the term $v_{ij} = (v_{j|i} + v_{i|j}) - v_{j|i}v_{i|j}$. The term v_{ij} is the local fuzzy simplicial set membership, based on the smooth nearest neighbors distances:

$$v_{j|i} = \exp\left[(-d(x_i, x_j) - \rho_i)/\sigma_i\right] \tag{2.8}$$

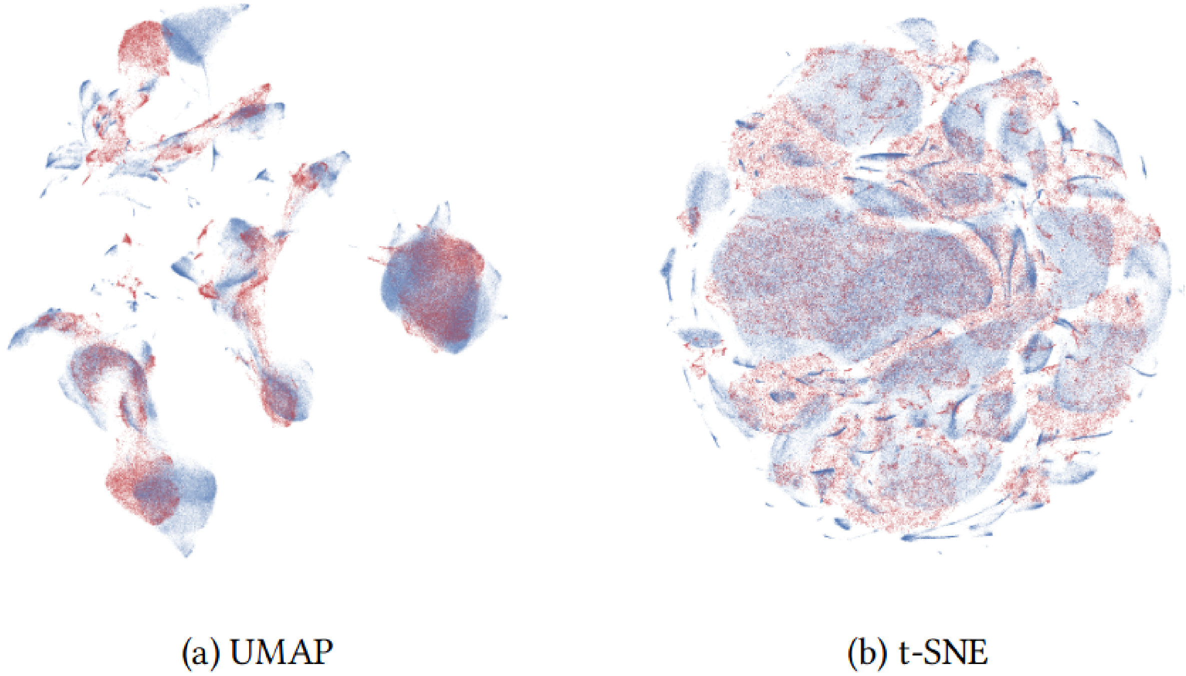

(a) UMAP (b) t-SNE

Figure 2.15: This figure illustrates the difference of performance between t-SNE and UMAP. Source: ©[103]

We call the low dimensional similarities :

$$w_{ij} = \left(1 + a\|y_i - y_j\|_2^{2b}\right)^{-1} \tag{2.9}$$

The overall understanding of the latent space representation is useful for synthesizing the interpretation of complex data structures. For our study, we observe PCA, t-SNE, and a UMAP as visualization techniques. We explore values in two-dimensional and three-dimensional graphs.

2.2.3 Singular Spectrum Analysis (SSA)

The Singular Spectrum Analysis was developed at the beginning of the geophysical task by Varadi *et al.*, 1999 [107] to detect oscillations in short and noisy time series. In this work, the authors offer a generalization of the process. This technique has been extended to different fields, multivariate data analysis, signal processing, and approximate projectors for time-series forecasting, Moskvina *et al.*, 2003 [108]. *"Singular spectrum analysis is a method of time-series analysis based on the singular value decomposition of an associated Hankel matrix." [108].* Undoubtedly, the application of these techniques holds promise within the domain of time-series analysis, particularly in the identification and accentuation of recurrent patterns. According to Zhu *et al.*, 2022 [109], the SSA encompasses 4 steps: embedding, decomposition, grouping, and reconstructions. (View an example of signal reconstruction in Figure 2.17.)
The vector, $x = \{x_n, n = 1, ..., N\}$ into r components, $x = \sum_{k=1}^{r} y^{(k)}$, information is rearranged,

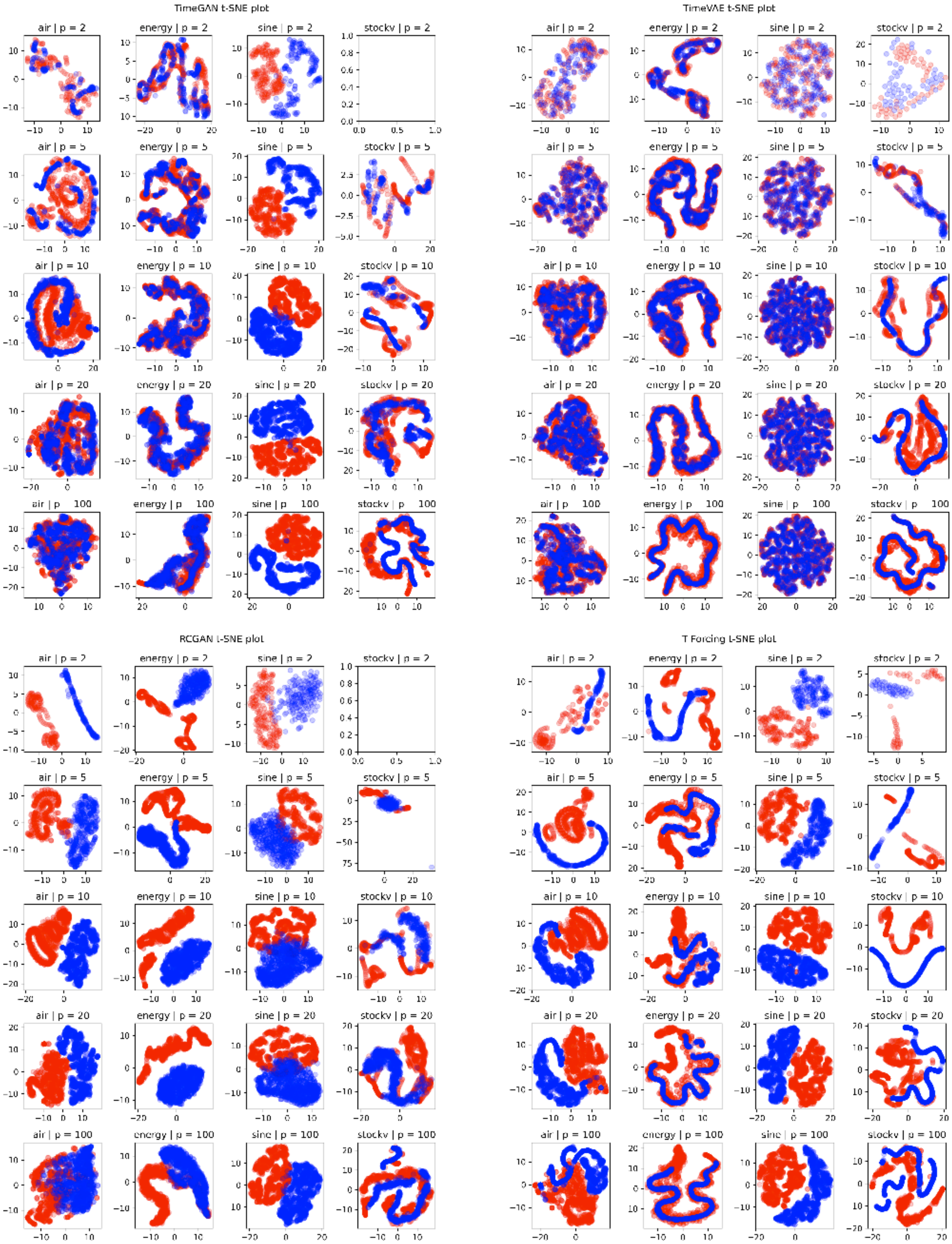


Figure 2.16: This t-SNE has been used here to analyze the architecture performance compared to others. In this paper, the author highlights that data generation performed through TimeVAE [104] is more realistic compared to generated data from other methods. t-SNE plots for the TimeGAN model proposed by ©[105] (top left), TimeVAE (top right), RCGAN [106] (bottom left), and T-Forcing (bottom right) models for the 4 data sets under various training size percentages (p). Red is the original data, and synthetic data is in blue. An empty t-SNE chart appears when insufficient training data are considered for the model to generate synthetic data.

such a way, we have the Hankel structure.

$$X_{L\times K} = \begin{pmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_L & x_{L+1} & \cdots & x_N \end{pmatrix} \triangleq \mathcal{H}[x] \tag{1}$$

A Singular Value Decomposition is operated :

$$U = [u_1, \dots, u_R]\,, \Sigma = diag\,(\sigma_1, \dots, \sigma_R)\,, V = [v_1, \dots, v_R] \tag{2}$$

. The energy distribution $\frac{\sigma_i^2}{\sum_{j=1}^{R} \sigma_j^2}$ of the i-th characteristic triple $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$ is called specturm of $\mathbf{x}$. In third place, we perform an Anti-diagonal Averaging

$$x_n^{(k)} = \frac{\sum_{|i+j|=n+1} X_{ij}^{(i)}}{\#\{(i,j)||i+j\,|=n+1\}} \tag{3}$$

. And Grouping :

$$x = \sum_{k=1}^{r} y^{(k)}, \text{ where } y^{(k)} = \sum_{i\in I_k} x^{(i)} \tag{4}$$

View the Algorithm 2.

Algorithm 2 SSA algorithm. Source [109]

```
**Input**: observed signals $\mathbf{x}$, embed dimension $L$
$U\Sigma V^\top \leftarrow \mathcal{H}(\mathbf{x})$
$\mathbf{x}_i \leftarrow \mathcal{H}^{-1}(\sigma_i, \mathbf{u}_i\mathbf{v}_i^\top)$
$I_k \leftarrow HCA(\mathbf{x}_u)$
$\mathbf{x}_k \leftarrow \sum_{i\in I_k} \mathbf{x}_i$
```

However, seismic signals have long time intervals. Using SSA necessarily implies a high computational load. The methodology becomes impractical. The main reason for this is the prohibitive computational demand.

2.3 Reconstruction and Generation

Most ML approaches in earthquake engineering treated seismic data using their 2D spectrograms, straightforwardly applying computer vision algorithms conceived for images. The database in seismology comprises time-series data in the form of a 3-channel component. Using 2D spectrograms introduces an additional pre-processing step, which can be streamlined by employing time-series data. However, more progress is needed for auto-regressive models in 1D time-series data. Furthermore, limited progress has been made in generative models, as evidenced by the sparse presentation of relevant papers in the work of Mousavi *et al.*, 2023 [18]. Therefore, a significant portion of our research focuses on addressing the challenges of reconstruction and the generation of time series. The reconstruction task is notoriously complex because of blur that affects the quality of the reconstructed samples (Bredell *et al.*, 2023 [110]). This aspect is presented in the section dedicated to auto-regressive encoder; see Section § 2.3.1. The approach pursued hereafter in this thesis involves a systematic exploration of the design

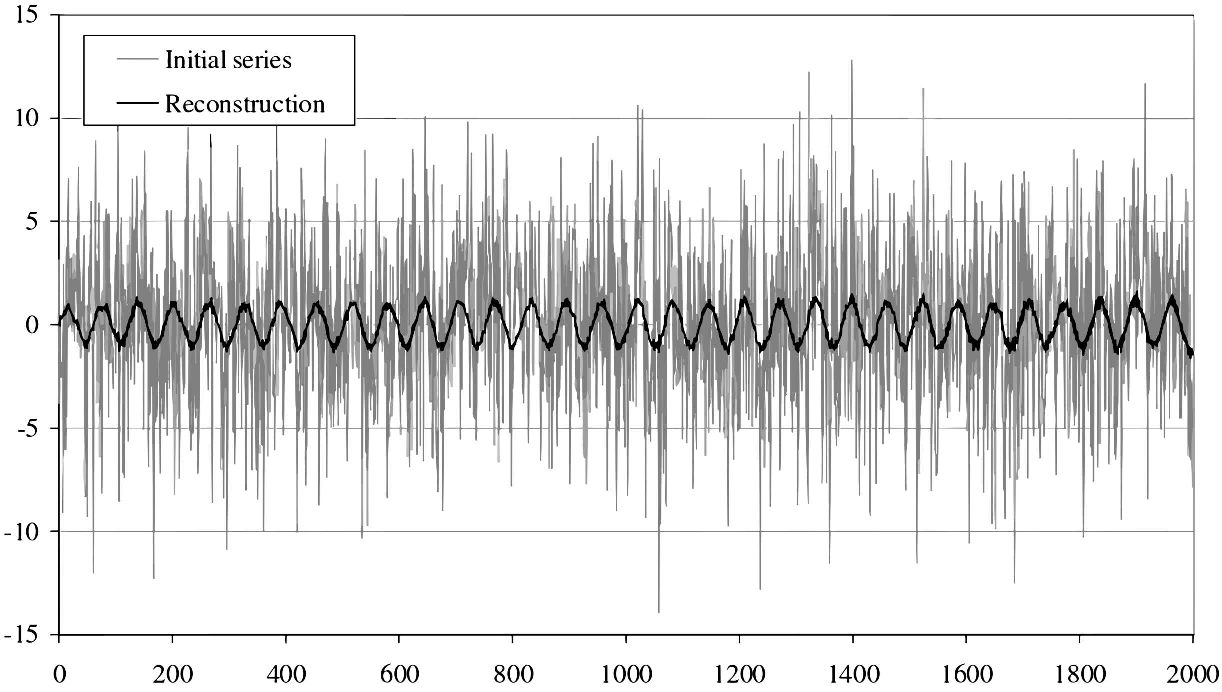


Figure 2.17: $\sin(0.1t) + e_t$, $e_t \sim \mathcal{N}(0, 16)$ and its reconstruction. Source:©Moskvina *et al.*, 2003 [108]

of neural network architecture adapted for this task. Subsequently, we learned how to design the neural network-adapted encoder and decoder architectures.

Additionally, we explored how latent representations (produced by the encoder) could impact the reconstruction, aiming simultaneously to identify the more convenient representation to generate new samples. The final purpose of the thesis is to provide a neural network architecture that enables multi-modal one-to-many generation of plausible broad-band time series conditioned by the same low-frequency input time series corresponding to different plausible "reconstructions". We propose a strategy to project data onto a Gaussian manifold (the latent representation). This aspect is explored in the chapter dedicated to translation, Chapter 4 and Chapter 5.

Consequently, we thoroughly examine all the models designed for the time-series data. The field of investigation spans simple autoencoders to more complex generative models. We treat the solution of Variational Auto Encoder (VAE, An *et al.*, 2015 [111]), Generative adversarial Network (GAN, Goodfellow *et al.*, 2014 [80]), Bi-directional GAN (BiGAN, Donahue *et al.*, 2017 [112]) and Contrastive Learning (CL, Chen *et al.*, 2020 [113]).

In the following sections, we present the advantage and the case of studies of this method in the field of seismology, as briefly depicted in Figure 2.18.

2.3.1 Auto-Encoder

An auto-encoder (AE) is a learnable mapping trained to match the identity. It reconstructs an input through an intermediate latent encoded representation and subsequently decodes it to reconstruct the input. Neural network architectures are often used as auto-encoders: the internal representation is derived by passing the inputted data through different layers, each down-sampling the original data while increasing the number of hidden features. The

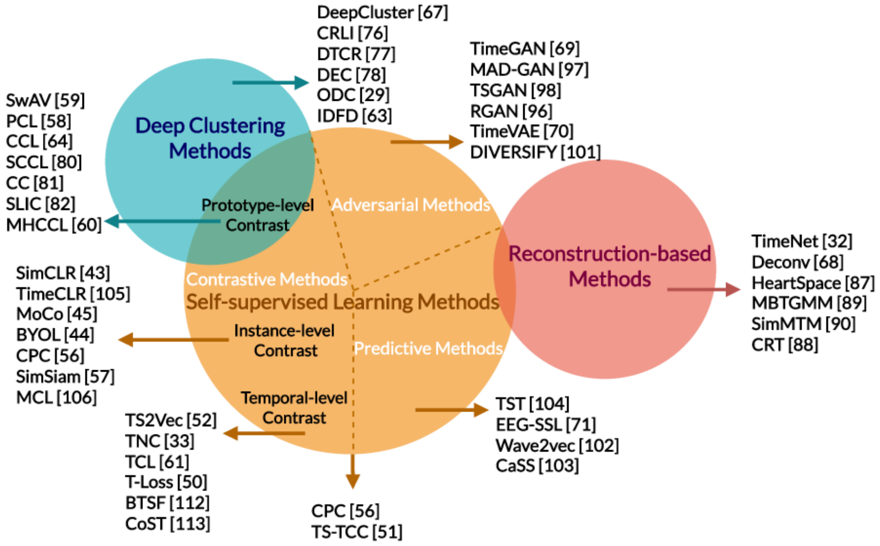


Figure 2.18: Unsupervised learning techniques for time-series. ©Illustration from Qianwen Meng *et al.*, 2023 [114]

latest layer that processes the down-sampled multiple features is often called the "bottleneck". Subsequently, another segment of the neural network processes these latent values through different layers, starting from abstract representation and passing it through different layers that up-sample it, with a final layer that should output the reconstructed data. In other words, to capture an abstract representation of data, we must encode the information in a space where we apprehend the relevant features. This latter will be used in a decoder to try to rebuild the same information. If we define $F : \mathcal{X} \to \mathcal{Z}$ the encoder, $G : \mathcal{Z} \to \mathcal{X}$ the decoder, and $\mathbf{x}$ the data we target to reconstruct, We could define the operation is as follows:

$$F : \mathbf{x} \mapsto \tilde{\mathbf{z}} = F(\mathbf{x}) \quad G : \mathbf{z} \mapsto \tilde{\mathbf{x}} = G(\mathbf{z}) \tag{2.10}$$

$\mathcal{Z}$ represents a latent manifold with a reduced dimension than the data space $\mathcal{X}$. To enforce the quality of reconstruction, different types of losses could be used, defined as a distance metric d between two samples:

$$L^p(\mathbf{x}, G(F(\mathbf{x})) = d(\mathbf{x}, G(F(\mathbf{x})) \tag{2.11}$$

Section 2.5 presents an in-depth discussion about loss minimization. In our project, we observed that integrating an auto-encoder represents a pertinent step in adversarial learning because it makes the network learn identity mapping in addition to the generation task.

Algorithm 3 AutoEncoder algorithm

```
Intialization of network paramters $\theta_g$
for $e$ in number of epochs do
    Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data $p_{\text{data}}(\boldsymbol{x})$
    Update the generator by ascending its stochastic gradient:
    $\eta_g \leftarrow \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{x}^{(i)} - G(F(\boldsymbol{x}^{(i)}))\|$
    $\theta_g \leftarrow \theta_g - \eta_g$
end for
The Adam algorithm is commonly used for that task
```

Application of Auto-Encoder

In the literature, the auto-encoder architecture is commonly used for :

(i) **Dimensionality reduction** : It has been observed that the auto-encoder has significant differences in dimension reduction when compared to commonly used methods such as PCA (Section § 2.2.1) and SSA (Section § 2.2.3). These differences demonstrate the limitations, which will be presented in the following paragraph.

PCA is based on a linear transformation to identify the principal components. Its application requires preliminary work before it can be applied to time series data. In addition, PCA operates in a space with an orthogonal direction that considers the most significant variances. In contrast, SSA, designed for time series, decomposes the data into additive components to identify the underlying patterns. AEs have demonstrated superior performance by outperforming conventional PCA and SSA (Helal *et al.*, 2017 [115], Casella *et al.*, 2011 [116], Roche et al., 2018 [117]). Autoencoders capture complex non-linear relationships within data because their architecture uses non-linear activation. Such an architecture could better observe the non-linearity in the data. Autoencoders are more flexible. They can be used for various tasks like anomaly detection, image compression, and feature engineering. In contrast, PCA is a more specialized tool better suited to data visualization and feature selection tasks. As a result, AE offers greater analytical power and better representation of data structures than the linear transformations of PCA and the additive decomposition of SSA.

(ii) **Feature extraction**: *"Features extraction is the process of transforming data into features in a way that is more tailored for analysis or modeling."*. Feature extraction has a widespread application, particularly in classification or regression. One of the many proposed applications is anomaly detection, Nazare *et al.*, 2018 [118]. The advantage of AE is the ability to extract relevant abstract information from data. Thus, AE highlights its inherent regularity properties. Section § 2.5 details feature extraction applications. In short, this capability improves the efficiency of feature extraction from data.

(iii) **Denoising**: This task, generally carried out by hand, is commonly called the "source separation problem". It involves isolating and attenuating instrument noise or background noise. In signal processing, the aim is to transform the signal in such a way as to preserve its quality and the intelligible information it contains effectively.

A time series is generally modeled as follows :

$$y(t) = x(t) * h(t) + n(t)$$

The formula where $y(t)$ is the distorted signal, $x(t)$ is the clean signal, $n(t)$ is the background noise, $*$ is a convolution operation, and $h(t)$ is a linear filter. Denoising can be seen as a source separation problem. The aim is to remove $n(t)$ and predict the desired speech, $\hat{x}(t)$ (Abdulatif *et al.*, 2022 [119]). In seismology, we are interested in acceleration data obtained from seismic waves. These data have the particularity of having a background noise. There are distinct phases that express the arrival of the P wave and then the arrival of the S wave. This reading enables the signal to be identified and denoised more effectively. It is, therefore, an effective way of reducing noise, Zhao *et al.*, 2023 [120]. This idea is also supported by Gulati *et al.*, 2020 [121].

Auto-encoders in engineering seismology

The first use of AE on seismic traces is from Valentin *et al.*, 2012 [122]. The authors have demonstrated how AE can be used to map from data to encoding space and its inverse. An illustration of this is shown in Figure 2.20. Seismic history matching is addressed by Mingliang Lu *et al.*, 2018 [123] with a Deep Convolutional Auto Encoder to sparsely represent the seismic data using latent features and then perform data assimilation. This research has been successfully used to transform multi-modal data into low-dimension space.
Loïc Viens *et al.*, 2019 [124] have proposed a convolutional autoencoder to de-noise ambient seismic field correlation functions. This method, noted ConvDeNoise, is designed to be applied to traditional two-station and single-station correlation functions to monitor the temporal evolution of the Earth's physical properties.

The ConvDeNoise shows that relative velocity computed on single station correlation denoise is better than classical correlations.
Autoencoder is used for Seismic facies recognition. Seismic facies classification is an important tool in sedimentary geology. It makes it possible to reconstruct the geological history of a region by identifying the different types of sediment present and the context in which they were deposited. The *"Prestack seismic data contains useful information that can help us find more complex atypical reservoirs."*. Unfortunately, this information has many redundancies that could block the use of classical mode. This issue is addressed by Qian *et al.*, 2021 [125] for seismic facies recognition. A convolution auto encoder's effectiveness is in detecting and removing redundancy that information as input could have. See Figure 2.19. As we can see, the different cases of using machine learning to manipulate autoencoders have interesting results for seismic. The effective field was for de-noising, feature extractors, and dimensionality reductions.

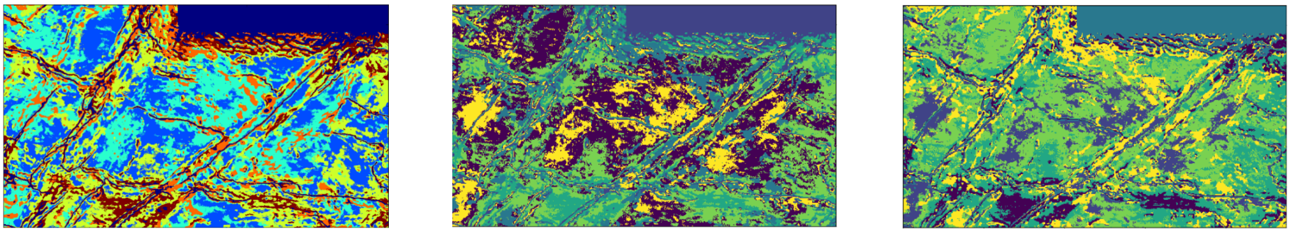
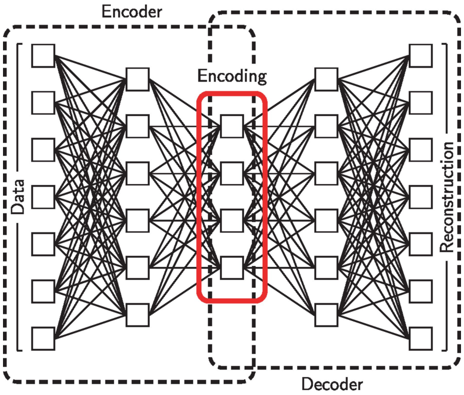


Figure 2.19: In this we present the Facies maps of LZB region. In (a) the result of using Convolutional AutoEncoder. In (b) the result using prestack data, in (c) the results using PCA based on prestack data. Source:ⓒQian *et al.*, 2021 [125]

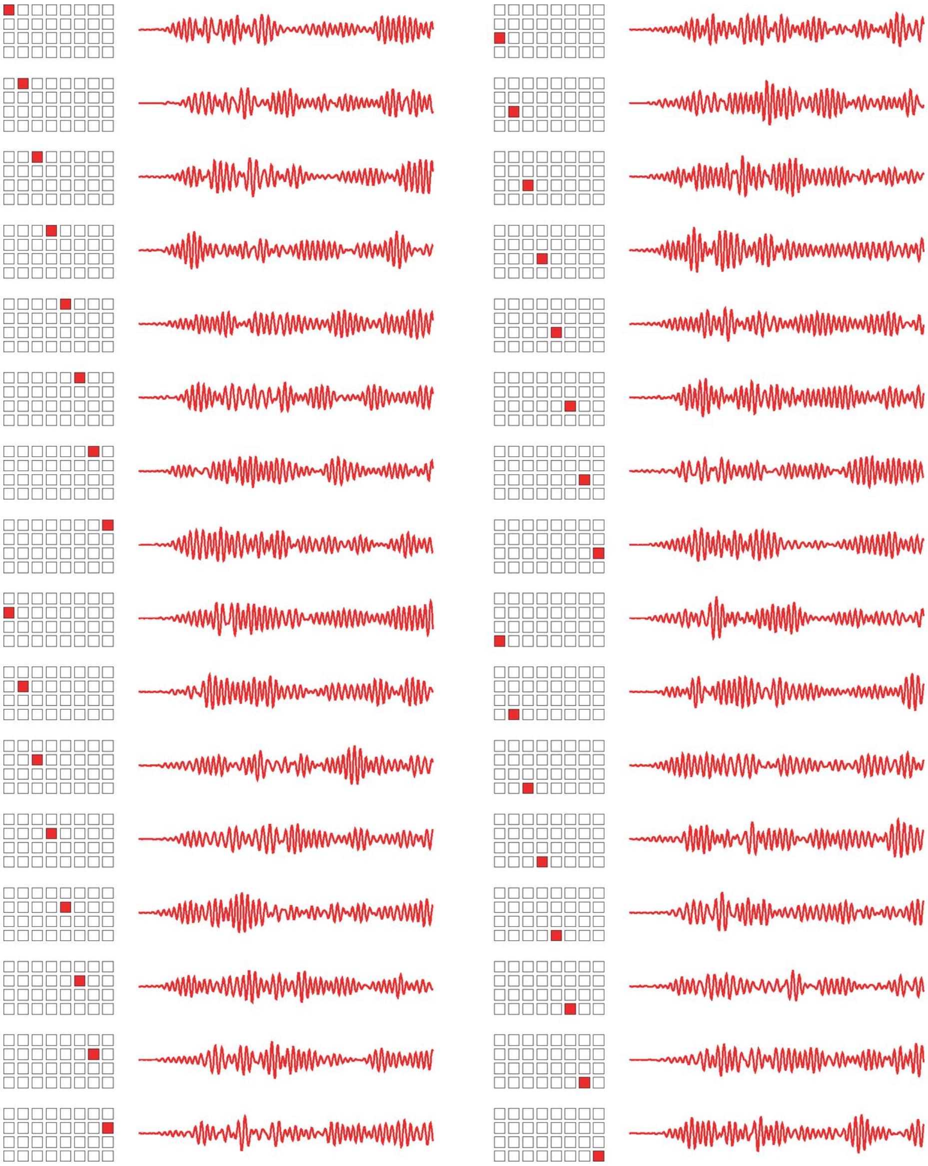
Limitations of the Auto-Encoder

(i) **Generalisation issues**: The architectural design of the AE heavily relies on the characteristics and intricacies of the data set. Consequently, it lacks a generalized approach. Such an approach cannot be universally applied across diverse data sets. This can hinder AE's ability to achieve robust and scalable solutions across different data sets. This is a generalization issue.

(ii) **Efficient separation of latent space**: The challenge inherent in AE lies in accurately disentangling the latent space. It is commonly observed that the latent representation may not effectively separate the underlying factors of variation in the data, Leeb *et al.*, 2021 [126]. This leads to no clear decantation of labeled data, for instance. In this sense, the regularization technique is further used to enhance the disentanglement capability expression, Cha *et al.*, 2022 [127].



(a) Architecture of an auto-encoder neural network, propose by Valenti *et al.*, 2012 [122]. Successive layers of neurons detect patterns in input data and use them to generate an encoded representation.



(b) Wave forms corresponding to unit encoding. Thirty-two wave forms obtained by decoding the unit vectors $\mathbf{x}(L/2) = (1, 0, \ldots, 0)$, $\mathbf{x}(L/2) = (0, 1, \ldots, 0)$, etc. We observe that each encoding element affects the entire length of the time-histories. Source [122]

Figure 2.20: Case of Auto-Encoder application to earthquake time-histories.

(iii) **Interpretability of the latent values**: Another notable limitation remains the interpretability of the representations learned by the AE, [126]. The latent space lacks clear and intuitive interpretations. Without interpretable results, it is difficult for human analysts to obtain meaningful information about the inner workings of the data projections.

In conclusion, while the Auto-Encoder method has demonstrated considerable utility in various domains, it is essential to acknowledge and address these limitations to leverage its full potential effectively.

Summary

An Auto-Encoder is composed of two parts. One is the encoder, F, and the other is the decoder, G. It could be unified or used in separate parts. In general the optimization functions is the ℓ_1 or ℓ_2 loss :

$$\|\mathbf{x} - G(F(\mathbf{x}))\| \tag{2.12}$$

This architecture is commonly used for dimensionality reduction, Feature extractions, and de-noising. However, this experience has some issues, especially with generalization, expressivity, and interpretability.

2.3.2 Variational Auto-Encoder

The variational autoencoder (VAE) could be seen as an additional step to the AE. VAE provides a better understanding of the latent representation of data. This representation is mathematically comprehensible. Often, the latent values of a sample are assumed to be a normal distribution with a mean of zero and a standard deviation of one unit.

Kingma and Welling [128] was the most ancient paper with Variational Auto Encoder, developed in 2014. The AE is a good platform for learning in directed probabilistic models. These models rely on latent variables, which are often hard to learn because of the difficulty of their posterior distributions. To overcome this challenge, variational autoencoders approximate the posterior distribution using neural networks, thus leading to efficient optimization and training of these models.

One key benefit of variational encoders is that they can handle continuous latent variables. MCMC sampling is among other inference methods that are not ideal for high-dimensional continuous latent variables since VAEs do not care about the dimensionality of the latent variables while approximating the posterior distribution using a neural network. This makes them especially useful when working on tasks involving heterogeneous data with high-dimensional hidden representations.

Let us define some data set $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, N represent the number of the samples. In VAE, the data are generated from some random processes, represented by $\mathbf{z}^{(i)}$; this latter is a supposed probability distribution, generated by some prior $p_\theta(\mathbf{z})$. Each sample is generated from some conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. Let design the model approaching the solution by $q_\phi(\mathbf{z})$.

The term $p(\mathbf{z})$ will be replaced by a model, $q(\mathbf{z})$ trained to parameterize a mathematical distribution, often chosen as Gaussian, Seeger *et al.*, 2011 [129]. The Gaussian Process is versatile and powerful, offering a natural generalization of multivariate random variables to infinite. It can capture complex data patterns and provide flexibility thanks to its non-parametric nature (Sfier *et al.*, 2022 [130]) simplicity and *interpretability*, smoothness and regularization, Bayesian learning and uncertainty quantification, adaptability and optimization, interpolation and Extrapolation of data, Rudolph *et al.*, 2023 [131].

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z};\theta)p_\theta(\mathbf{z})\mathrm{d}\mathbf{z} \tag{2.13}$$

The term $\epsilon \sim N(0, I)$ the term $z = \mu(X) + \Sigma^{1/2}(X) \cdot \epsilon$. an ELBO (Evidence Lower Bound) is the solution to satisfy:

$$\log p_\theta(\mathbf{x}) - \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) = E_{\mathbf{z}\sim q_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) \tag{2.14}$$

The term $\mathcal{D}_{KL}(\cdot)$ introduced here is the Kullback Leibler distance. The complete function to minimize combines the auto-encoder loss and the Kullback Leibler loss. In the VAE, it is assumed that the distribution of the latent space we want to map is a Gaussian distribution of mean 0 and standard deviation 1; See Figure 2.21. The latter leads to the following equations:

$$\begin{aligned} \mathcal{D}_{KL}[\mathcal{N}(\mu(\mathbf{x}^{(i)}), \Sigma(\mathbf{x}^{(i)})) \| \mathcal{N}(0, \mathbf{I})] = \frac{1}{2}(&\mathrm{tr}(\Sigma(\mathbf{x}^{(i)})) + (\mu(\mathbf{x}^{(i)}))^\top(\mu(\mathbf{x}^{(i)})) \\ &- k - \log\det(\Sigma(\mathbf{x}^{(i)}))) \end{aligned} \tag{2.15}$$

Then the cost functions to minimize is :

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right) = -D_{KL}\left(q_{\boldsymbol{\phi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| p_{\boldsymbol{\theta}}(\mathbf{z})\right) + \mathbb{E}_{q_{\boldsymbol{\phi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right] \tag{2.16}$$

In general, we approach the solution using neural network architecture. Let us define G as the decoder and F as the encoder. The weak formulation of the previous Equation 2.16 is then :

$$\mathcal{L}(G, F; \mathbf{x}^{(i)}) = L^p(\mathbf{x}^{(i)}, G(F(\mathbf{x}^{(i)})) + \mathcal{D}_{KL}[\mathcal{N}(\mu(F(\mathbf{x}^{(i)}), \Sigma(F(\mathbf{x}^{(i)})) \| \mathcal{N}(0, \mathbf{I})] \tag{2.17}$$

Algorithm 4 Variational AutoEncoder algorithm

```
Intialization of network paramters $\theta_g$
for $e$ in number of epochs do
    Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data $p_{\text{data}}(\boldsymbol{x})$
    Calculating $\mu(X)$ and $\sigma(X)$:
    $\mu(\boldsymbol{x}^{(i)}), \sigma(\boldsymbol{x}^{(i)}) \leftarrow F(\boldsymbol{x}^{(i)})$
    Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
    $\boldsymbol{z}^{(i)} = \mu(\boldsymbol{x}^{(i)}) + \sigma(\boldsymbol{x}^{(i)}) \odot \boldsymbol{\epsilon}$
    Update the generator by ascending its stochastic gradient:
    $\eta_g \leftarrow \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{x}^{(i)} - G(F(\boldsymbol{x}^{(i)}))\| + \mathcal{D}(\mu(\boldsymbol{x}^{(i)}), \sigma(\boldsymbol{x}^{(i)}), 0, 1)$
    $\theta_g \leftarrow \theta_g - \eta_g$
end for
The Adam algorithm is commonly used for that task
```

VAE in time-series Seismology

(i) Mingliang Liu et al., in their work published in 2022 [132], proposed the use of a Variational Autoencoder (VAE) to quantify stochastic inverse problems. Their study focuses on estimating rock and fluid properties in the subsurface from geophysical measurements,

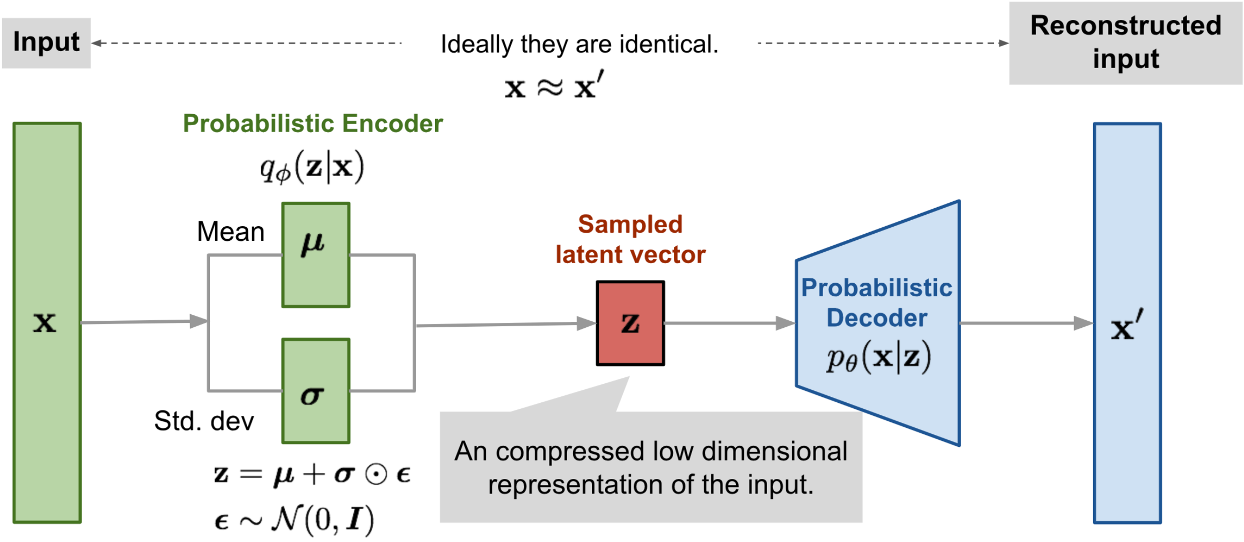


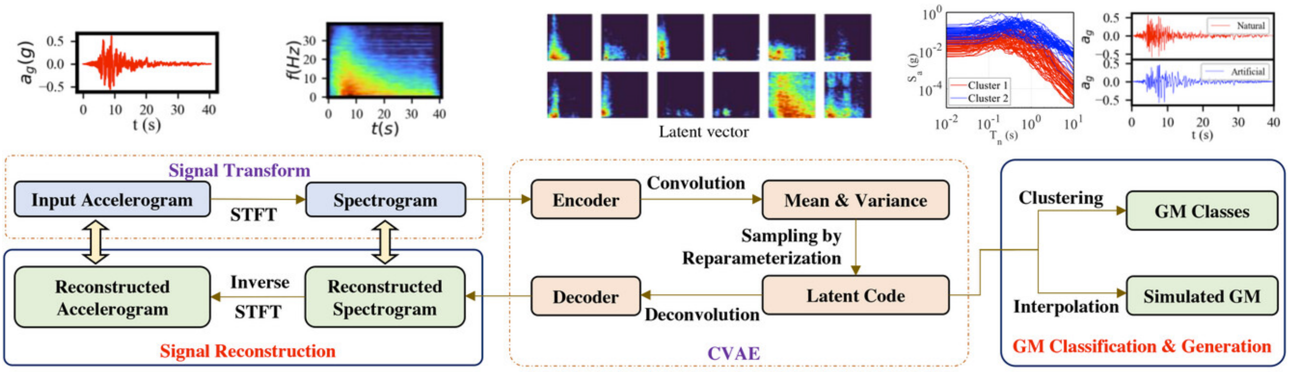
Figure 2.21: Illustration of Variational Auto Encoder to reconstruct data. Source: lilianweng

a computationally and memory-intensive process. In scenarios where dimensionality reduction is preferred as a strategy due to the high computational demands, they highlight that reduction based on deep generative models may lead to information losses, affecting the quality of reconstructed data.
To address this issue, the authors introduce a workflow that compares Markov Chain Monte Carlo (MCMC) results with latent values generated by the VAE. This framework is specifically applied to geophysical inverse problems. The findings of their study indicate that employing dimensionality reduction, in this case using VAE, can lead to more efficient solutions for inverse stochastic problems.

(ii) Ning *et al.*, 2024 [133] use Convolutional VAE to enhance Ground motion analysis and achieve reliable GM classification, selection, and generation of simulated motions. See Figure 2.22.

(iii) Haowei Hua proposes GMVAE to address the problem of seismic facies deep clustering, Hua *et al.*, 2O23 [134]. The problem of Seismic Facies Analysis (SFA) could be a crucial step in interpreting subsurface structures. The fact that the problem remains intractable by the classical algorithm has led Hua et al. to develop a novel framework, an unsupervised end-to-end-based SFA method, LMVAE (lognormal mixture-based variational autoencoder) and enhance the existing GMVAE (Gaussian mixture variational autoencoder-based) framework. See Figure 2.23.

(iv) TimeVAE [104] The Time Variational Autoencoder (Time VAE) is trained using the Kullback-Leibler loss and follows the Evidence Lower Bound loss (ELBO) principle. Time VAE employs a dedicated decoder in this framework that incorporates assumed target distributions, often Gaussian distributions. This approach provides a versatile method for data generation. Importantly, TimeVAE introduces temporal structure into the data generation process within the decoder, enhancing our ability to interpret and understand how the data is generated.



(a) overall framework

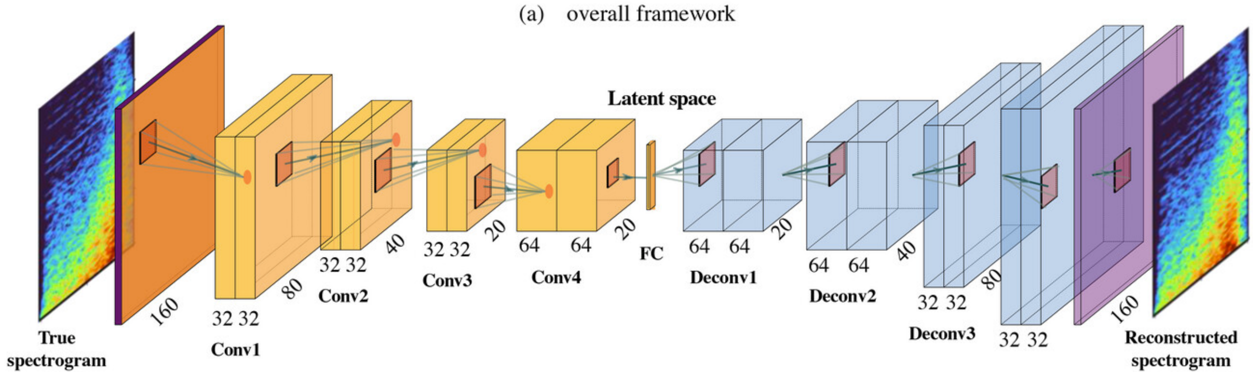


Figure 2.22: *"Proposed short-time Fourier transform (STFT) and convolutional neural network–variational autoencoder (CNN-VAE) end-to-end framework for ground motion (GM) classification and generation."*, Source ©Ning *et al.*, 2024 [133]

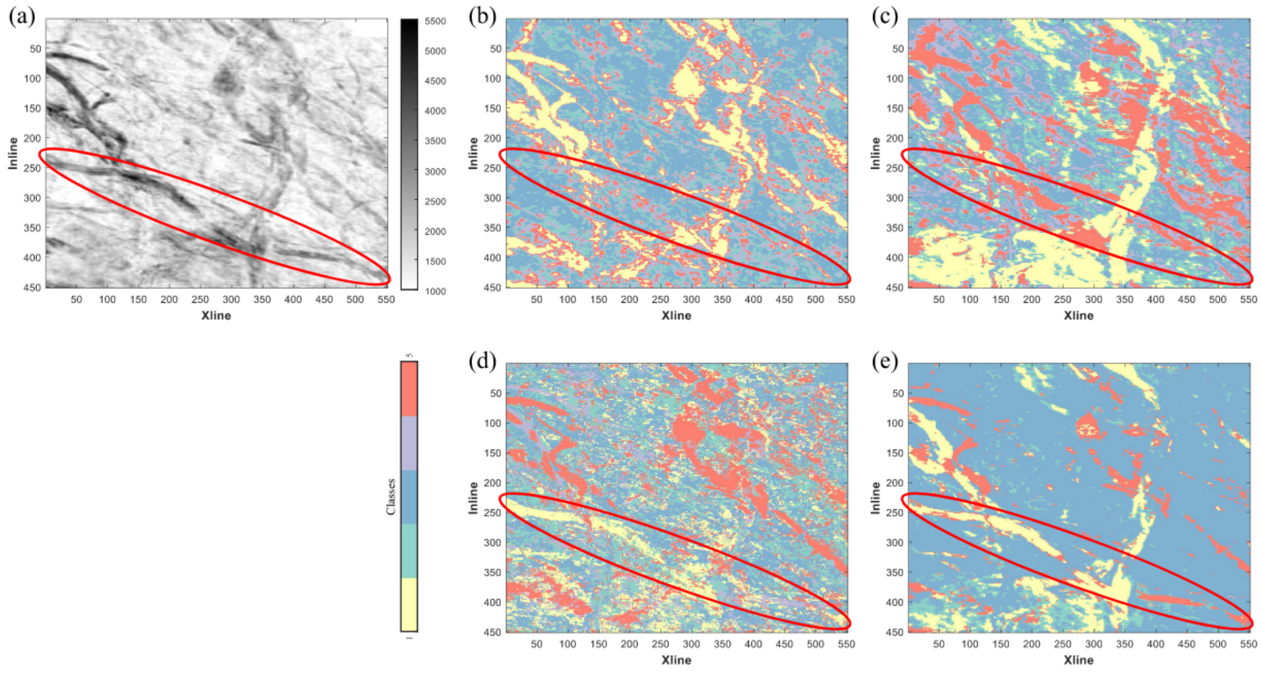


Figure 2.23: "SFA results of Zhongjiang field data. (a) RMS amplitude attribute of field data. (b) Gabor+SOM. (c) DCAE+SOM. (d) GMVAE. (e) LMVAE.". Source: Hua *et al.*, 2023 [134].

Application of Variational AutoEncoder

The VAE is used for different tasks.

(i) **Latent space regularization** Observation of the latent values of the Encoder could be done through t-SNE. We observe that adjacent data points within the same latent space are observed to correlate with analogous data samples in the original data set. Consequently, the integration of algebraic operations within these latent spaces becomes conceivable, facilitating the interpolation between points. This interpolation process is orchestrated to encapsulate sought-after or similar outcomes. Notably, the attributes extracted from this latent space process have significance and explicable characteristics. This produced research enables the trajectory particles to align with the predefined data sample.

(ii) **Continuous Latent Space**. Unlike a configuration of distinct points, this type of latent space is continuous. This means that similar data have latent representations close to each other. Hence, the generation smoothly changes. Data representation is compact (See the illustration in Figure 2.24). The interpolation is possible.

$$\mathbf{z} = \mu(\mathbf{x}) + \sigma(\mathbf{x}) \odot \epsilon \tag{2.18}$$

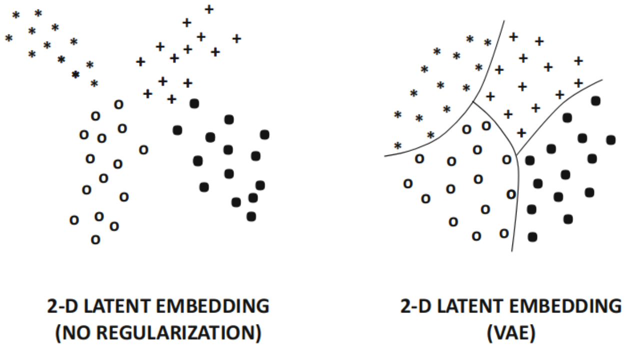


Figure 2.24: Compact representation of Latent space. ©[135]

(iii) **Variational Inference** VAE is close to Generative Adversarial Network (GAN, view Section § 2.3.3). Through Gaussian, it could produce meaningful outputs. Further investigation has proved that generation respects the data distribution.

VAE represents a paradigm change compared to previous architecture. In addition to providing reconstruction and meaningful representation, it is a regularizer for latent space. This framework allows a smooth transition for these latent values. Finally, it could be played for Variational Inference.

Limitations of the Variational Encoder

The efficacy of the Variational Auto Encoder, while exhibiting advancements beyond the conventional Auto Encoder technique, is accompanied by inherent limitations within its scope of application.

It is well known that the Variational Auto Encoder (VAE) needs to give accurate reconstruction. The expected reconstructed signals appear to be either noisy or typically a noise. The nature of the loss used is the main reason for that issue.

Moreover, mode collapse is one of the common issues of this strategy, and we need many robust solutions for that issue. The model could not capture diversity through its distribution. Only a few samples are reproduced in that case. Furthermore, the VAE struggles with large-scale data sets.

Summary

Variational AutoEncoder (VAE) introduces controlled representation for data. Unlike a conventional autoencoder, the VAE provides better interpretability of latent space data. The equation to minimize is :

$$\mathcal{L}(G, F; \mathbf{x}^{(i)}) = L^p(\mathbf{x}^{(i)}, G(F(\mathbf{x}^{(i)})) + \mathcal{D}_{KL}[\mathcal{N}(\mu(F(\mathbf{x}^{(i)}), \Sigma(F(\mathbf{x}^{(i)}))\|\mathcal{N}(0, \mathbf{I})]$$

VAE is used in seismology to estimate rock and fluid properties in the subsurface from geophysical measurements, improve ground motion analysis, group seismic facies at depth, and generate synthetic seismic data. It is also applied to time series data. Despite the various advantages offered by this framework, VAE needs help with a severe problem: either the reconstruction quality is noisy, or the model fails to reproduce the diversity of the database. For these reasons, new approaches are preferred.

2.3.3 Generative Adversarial Network

Goodfellow and his collaborator proposed the concept of a Generative Adversarial Network for the first time in 2014 [80]. GAN outperforms VAE in generating new samples because the latter needs better reconstruction and blurry generation, and it is hard to upscale to a vast database. GAN is trained according to the adversarial learning framework, a relevant case study of the Nash Equilibrium of a min-max game. This framework introduces two agents: a Generator, G, and a Discriminator, D (represented by two neural networks). The equilibrium is satisfied when the discriminator cannot distinguish what is the difference between the real and the generated values. To steer D and G towards the equilibrium, the parametric distribution associated with the data generated by G is tuned to approach the original data distribution, minimizing the Jensen-Shannon distance between the two (or, in a variant, the Earth's mover distance). G transforms a noise sampled from a prior arbitrary probability distribution, Gaussian or Uniform, into a generated data sample. The mini-max game can be formulated according to the following expression:

$$\begin{aligned} \min_G \max_D V(D, G) = & \quad \mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \\ = & \int q(\boldsymbol{x}) \log(D(\boldsymbol{x})) d\boldsymbol{x} + \int p(\boldsymbol{z}) \log(1 - D(G(\boldsymbol{z}))) d\boldsymbol{z} \end{aligned} \tag{2.19}$$

The training of G is generally performed by minimizing $\log(D(G(\mathbf{z})))$ instead of the second term in Equation 2.19 (see the proof in [136]). However, when the discriminator improves its capability of discerning real data from the generated one, the generator's gradient should vanish. Under these assumptions :

(i) the optimal values $\|D - D^*\| < \epsilon$

(ii) the gradient of the discriminator goes to zeros : $\|\nabla D^*(G(z))\| = 0$

(iii) The Jacobian of the generator is bounded : $J_\theta(G(z)) \leq M^2$

it holds that the gradient of second term the in Equation 2.19 is bounded as follows:

$$\left\|\nabla_\theta \mathbb{E}_{z\sim p(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right]\right\|_2 \leq \mathbb{E}_{z\sim p(z)}\left[\frac{\epsilon^2 \left\|J_\theta G(z)\right\|_2^2}{(1-\epsilon)^2}\right] = M\frac{\epsilon}{1-\epsilon}$$

So

$$\lim_{\|D-D^*\|\to 0} \nabla_\theta \mathbb{E}_{z\sim p(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] = 0$$

"The fact that this happens is terrible since whether the generator's cost function is close to the Jensen-Shannon divergence depends on the quality of this approximation. This brings us to a fundamental point: either our discriminator updates will be inaccurate or disappear. It is, therefore, difficult to train using this cost function or leave it to the user to decide the precise amount of training devoted to the discriminator, making G.A.N. training extremely difficult.", According to Martin Arjovsky, *et al.*, 2017 [136].

This is the reason it is recommended to change the gradient of the Generator to :

$$\nabla_\theta \mathbb{E}_{z\sim p}\left[-\log\left(D\left(G(z)\right)\right)\right] = \Delta\theta$$

So the optimal values for $D* = \frac{P_r}{p_G+P_r}$ should be :

$$\mathbb{E}_{z\sim p(z)}\left[-\nabla_\theta \log D^*\left(G(z)\right)|_{\theta=\theta_0}\right] = \nabla_\theta\left[D_{\mathrm{KL}}\left(\mathbb{P}_G\|\mathbb{P}_r\right) - 2D_{\mathrm{JS}}\left(\mathbb{P}_G\|\mathbb{P}_r\right)\right]|_{\theta=\theta_0}$$

The global minimum for the learning $C(G)$ is achieved if and only if $p_G = p_{data}$, So $C(G) = -\log 4$.

Algorithm 5 The Generative Adversarial Network algorithm

```
Intializatio of network paramters $\theta_d$, $\theta_g$
for $e$ in number of epochs do
    for $k$ in number of discriminator training step do
        Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from prior $p_g(\boldsymbol{z})$
        Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data $p_{\text{data}}(\boldsymbol{x})$
        Update the discriminator by ascending its stochastic gradient:
        $\eta_d \leftarrow \nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D(\boldsymbol{x}^{(i)}) + \log(1 - D(G(\boldsymbol{z}^{(i)})))\right]$
        $\theta_d \leftarrow \theta_d - \eta_d$
    end for
    Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from prior $p_g(\boldsymbol{z})$
    Update the generator by ascending its stochastic gradient:
    $\eta_g \leftarrow \nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log D(G(\boldsymbol{z}^{(i)}))$
    $\theta_g \leftarrow \theta_g - \eta_g$
end for
The Adam algorithm is commonly used for that task
```

GAN for time-series

(i) MAD-GAN [137] (Multivariate et al. for Time Series Data with Generative Adversarial Networks) is one of the earliest standard GAN models for anomaly detection in multivariate time series. Both generator and discriminators are deep neural networks based on LSTM and RNN blocks (Long *et al.* cells and Recurrent Neural Networks, see Figure 2.25).

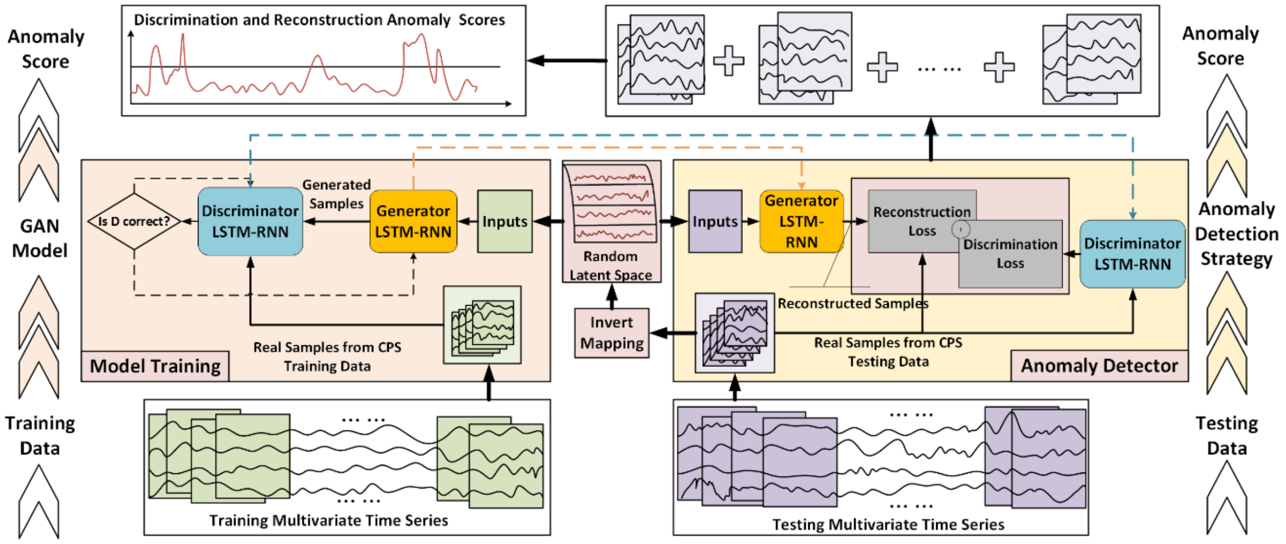


Figure 2.25: MAD-GAN: Unsupervised GAN-based anomaly detection. On the left is a GAN framework in which the generator and discriminator are obtained with iterative adversarial training. On the right is the anomaly detection process, where a trained discriminator and generator are employed to compute a combined anomaly score based on discrimination and reconstruction. Source : ©Li *et al.*, 2019 [137]

First, while the training set is used to train the Generator G_{RNN} and the Discriminator D_{RNN} in a classical GAN fashion, the test data set is used for anomaly detection only. In this sense, the Adversarial Loss satisfied is :

$$\begin{aligned}\min_{G_{RNN}} \max_{D_{RNN}} V(D_{RNN}, G_{RNN}) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D_{RNN}(\boldsymbol{x})] \\ + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D_{RNN}(G_{RNN}(\boldsymbol{z})))]\end{aligned} \tag{2.20}$$

Secondly, The authors use the trained G_{RNN} and D_{RNN} to detect anomalies in the test set, $\mathcal{X}^{test}$. Instead of projecting the data into a latent space, MAD-GAN adopts an anomaly score for discrimination and reconstruction (performed with standard autoencoder loss). Since fake samples are generated based on random latent noise, the test data, $\{\mathbf{x}_i\}^{test}$ is searched through a random generation of Gaussian variables $\mathbf{z}_k$. This closest sampled Gaussian noise is less dissimilar to the test sample. At the moment the similarity is proven by the loss, this noise above,$\mathbf{z}^k$, is selected as the one in correspondence with the $\mathbf{x}_i^{test}$, in the test set. The procedure is how the authors explicitly map data and its Gaussian projection, with the error function $E_r(\cdot)$ defined as :

$$\begin{aligned}\tilde{\mathbf{z}}^{(k)} &= \min_{\mathbf{z}^{(i)}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} E_r\left(\mathbf{x}^{(i),test}, G_{RNN}(\mathbf{z}^{(k)})\right) \\ &= \min_{\mathbf{z}^{(i)}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \left(1 - \text{sim}(\mathbf{x}^{(i),test}, G_{RNN}(\mathbf{z}^{(i)})\right)\end{aligned} \tag{2.21}$$

Formula in which, $\text{sim}(\mathbf{u},\mathbf{v}) = \mathbf{u}^\top\mathbf{v}/(\|\mathbf{u}\| \cdot \|\mathbf{v}\|)$ performs as a cosine. Finally, $\mathbf{z}^k$ is considered as corresponding mapping to the $\mathbf{x}^{test}$. In this sense, the authors have built the corresponding couple $(\mathbf{x}^{test,i}, \tilde{\mathbf{Z}}^k)$, where k is the component of the sample $\mathbf{z}$, closest to $\mathbf{x}^{i,test}$ of the test set. Moreover, the authors introduce another formulation for identifying anomalies in data. This quantity is called the residual, calculated at the time step t as:

$$\text{Res}(\mathbf{x}_t^{test}) = \sum_{i=1}^{n} \|\mathbf{x}_t^{test,i} - G_{RNN}(\tilde{\mathbf{z}}_t^{k,i})\| \tag{2.22}$$

An anomaly loss is therefore created based on the combination of the probability of Discriminators and the Residual values.

$$L_t^{test} = \lambda \mathrm{Res}(\mathbf{x}_t^{test}) + (1-\lambda) D_{RNN}(\mathbf{x}_t^{test}) \tag{2.23}$$

The Discrimination and Reconstruction Anomaly Score, known as the DRS score, maps the anomaly detection loss of sub-sequences against the original time series.

$$\begin{aligned} DRS_t &= \frac{\sum_{j,s\in\{j+s=t\}} L_{i,s}}{l_{ct}} \\ l_{ct} &= \mathrm{count}(j, s \in \{j+s=t\}) \end{aligned} \tag{2.24}$$

The value $s \in \{1, 2, \cdots, s_w\}$. Where s_w is the window size. The Maximum Mean Discrepancy MMD is also used to evaluate whether the GAN model has learned the training data distributions.

$$\begin{aligned} MMD^2(X,Y) = &\frac{1}{m(m-1)} \sum_i \sum_{j\neq i} k(\mathbf{x}_i, \mathbf{x}_j) \\ &- 2\frac{1}{m.m} \sum_i \sum_j k(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{m(m-1)} \sum_i \sum_{j\neq i} k(\mathbf{y}_i, \mathbf{y}_j) \end{aligned} \tag{2.25}$$

Where :

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) = \exp\left(\frac{-1}{\sigma^2}[\mathbf{x_i}^{\intercal}\mathbf{x}_i - 2\mathbf{x}_i^{\intercal}\mathbf{x}_j + \mathbf{x}_j^{\intercal}\mathbf{x}_j]\right)$$

Finally, different time series data sets have been used to prove the effectiveness of the MAD-GAN. To sum up, this process recursively involves :

(a) The training and the update of G_{RNN} and D_{RNN} weights, through back-propagation of the adversarial loss. Equation 2.20.

(b) Proceed to the mapping testing data back to latent space: Use trained G_{RNN} with random noise to make a collection mapping with the test set, $(\mathbf{x}_i^{test}, G_{RNN}(\mathbf{z}_k))$, where $G_{RNN}(\mathbf{z}_k) \approx \mathbf{x}_i^{test}$.

(c) Calculate the residual loss and update

(d) Calculate the discrimination results, $D_{RNN}(\mathbf{x}^{test})$

(e) obtain, finally, the anomaly score with DRS_t

(ii) TSGAN, Smith *et al.*, 2020 [138], i.e., time-series GAN, surpasses another method for the generation of realistic time-series sequences as output, particularly where data acquisition is limited. The authors employ two types of GAN, one unconditionally trained and the second conditionally trained. See Figure 2.26. The unconditionally-trained GAN generates synthetic time series, which are then improved by the conditionally-trained one. The first GAN is used to generate a 2D spectrogram from random noise. The synthetic data generated, with the generator G, by the first GAN are considered as essentially synthetic power spectral densities of the time series data. The loss satisfied here is:

$$L_1 = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[D_x(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[D(G(z))] + \lambda \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}))\|_2 - 1)^2] \tag{2.26}$$

Subsequently, the second GAN, with the generator F, uses the 2D spectrogram coming from $G(z)$ as an entry and outputs a time-series data $F(G(z))$. Such a process satisfies the loss below :

$$L_2 = \mathbb{E}_{\mathbf{y}\sim\mathbb{P}_r}[D_y(\mathbf{y})] - \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[D(F(G(\mathbf{z})))] + \lambda\mathbb{E}_{\hat{y}\sim\mathbb{P}_{\hat{y}}}[(\|\nabla_{\hat{\mathbf{y}}}D(\hat{\mathbf{y}}))\|_2 - 1)^2] \tag{2.27}$$

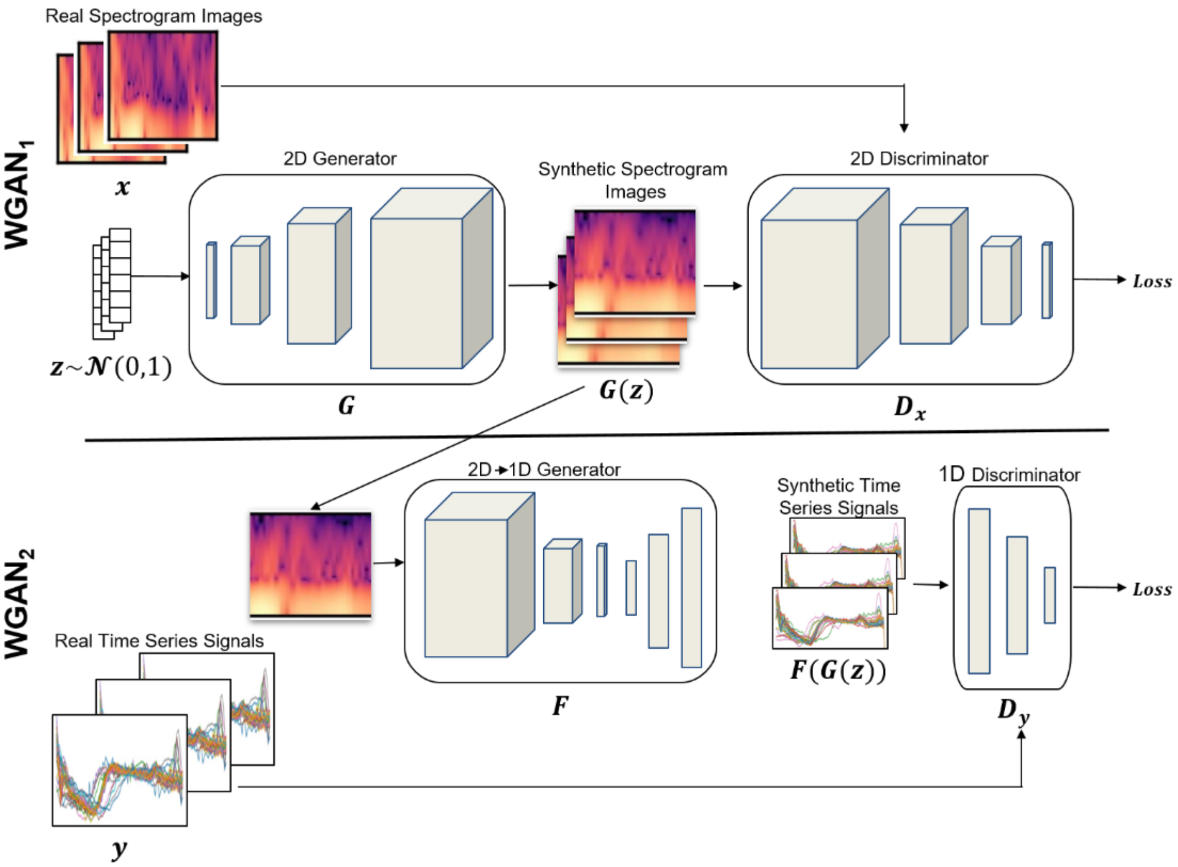


Figure 2.26: TSGAN framework from [138], ©Smith *et al.*, 2020

(iii) RGAN, also named Recurrent GAN and Recurrent Conditional GAN (RCGAN) developed by Esteban *et al.*, 2017 [139] are developed to output Real-valued multi-dimensional time series. The Generator and the discriminator are formed with a Recurrent neural network. The standard GAN loss is used for that task. The authors employed the maximum mean discrepancy (MMD) to evaluate whether the generator captured the data distribution.

GAN in Seismology

In seismology, GAN was applied around 2020. Florez e*t al.*, 2022, [88]. they have proved that the technique could generate ground motion conditioned on continuous physical variables (magnitude, epicentral distance, $V_{S,30}$). The GAN is trained over 260,764 earthquake time series, recorded at the free surface, mainly in Japan (K-NET and KiK-Net). The case study includes earthquakes with magnitudes from 4.0 to 7.5 and stations at an epicentral distance between 0 and 280 km. The 100 Hz earthquake is downsampled to 20 Hz; a 20-second window is selected afterward. Values of the signal are normalized and encompassed between [-1, 1]. The maximum Peak Ground Acceleration in their data set is $1.2g$. The GAN generates realistic earthquake time histories with meaningful conditioning where no earthquake has been recorded. Nevertheless, this does not represent a one-to-many mapping; only visual considerations can be made. Nothing even guarantees that the quality of the generation is relevant. See Figure 2.27. Although similar to our research, the study by Florez *et al.* 2022 is based on relative acceleration data collected from earthquakes in Japan. In this thesis, we will focus on data available in the STEAD database, a collection of seismic recording data distributed over the entire surface of the globe. Moreover, the generation process in the work of Florez *et al*, incorporates Gaussian noise and metadata. Our research will focus on generating synthetic earthquake signals. This time, we will use low-frequency signal encoding to constrain the output. A Gaussian distribution is also used in our model. More than guarantee a one-to-one mapping, our study pushes the step further until we obtain a one-to-many mapping. Results are physically constrained to respect the arrival time of the P- and S-waves and the frequency coherence with the targeted seismic signal. Yuanming Li et al., 2020 [140] proposed a GAN featuring a Gated convolutional neural network playing the role of the Generator to capture better the relevant structure of the time series of a ground-motion signal. This is made to improve earthquake detection and classification. The training data set is provided by KMA (Korea Meteorological Administration), covering the period from January 2016 to July 2018 with a magnitude range above 3.0, recorded at 256 locations minimum. Each sample is a 3-component time-history, sampled at 100 Hz. Despite the interest of such a Gated CNN, generated data were quite similar to each other, making the generation ability less relevant.

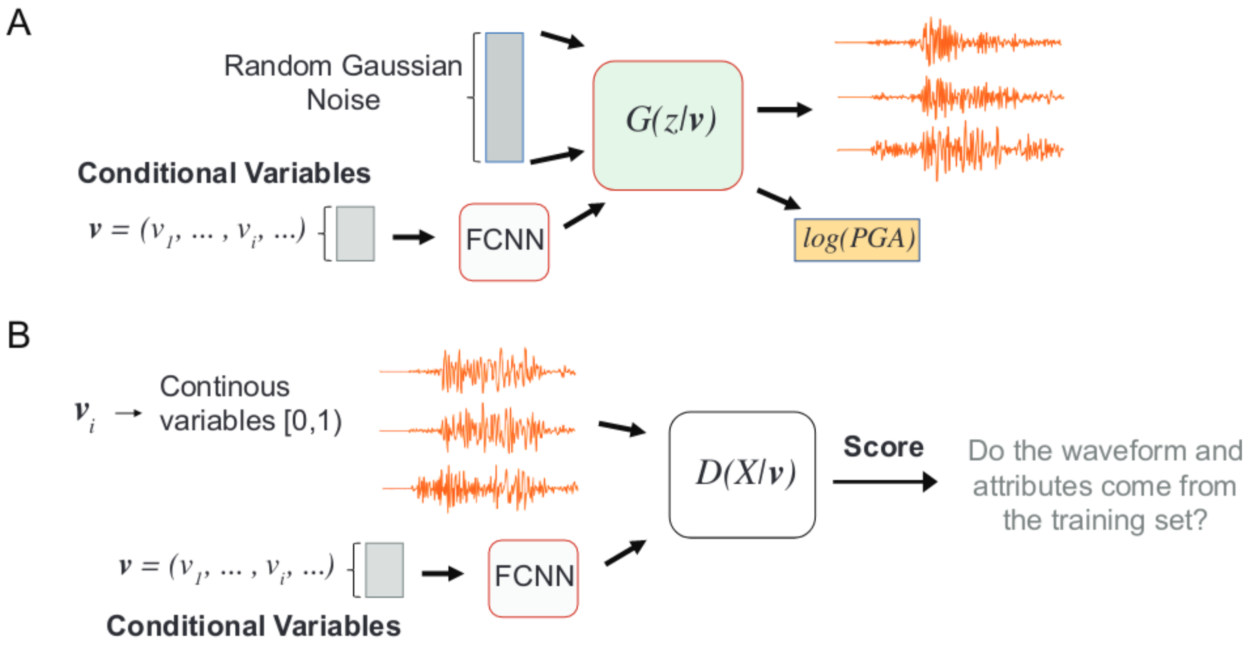


Figure 2.27: Model developed by ©Florez *et al.*, 2022 collaborators [88]. In (A), it is the Conditional Generation Model, and in (B), the conditional Discriminator is used.

Furthermore, we do not find any benchmarks or comparisons for different types of architecture. Particularly in evaluating the improvement of the gated convolution compared to classical

Subject	References	Model	Applications
Earthquake Engineering	Li *et al.* (2018) [141]	GAN+Ref CNN	Feature extractor and Random Forest classifier for EEW system on P-waves
EEW	Kim and Torbo (2019) [142]	WGAN-GP	Synthetic seismic data generation for EEW system
	Meier *et al.* (2019) [143]	GAN+ RF	Five DL models comparison for EEW
	Wu *et al.* (2021) [82]	EQGAN	Seismic data augmentation for EEW for low-cost MEMS sensors
	Liu *et al.* (2022a, 2022b) [144], [145]	GAN	Seismic data discrimination between earthquakes and microtremors
	Li *et al.* (2021a) [146]	GAN-LSTM	Discrimination and recognition of seismic array noise and signal
Augmentation	Wang *et al.* (2019a, 2021) [147], [148] , Li et al. (2020a) [149]	cGAN	Generate synthetic seismic waveforms for data augmentation
	Li *et al.* (2020b) [84]	GAN	Seismic time series signals synthesis with gated CNN
	Gatti and Clouteau (2020) [21]	WGAN	Physics-based simulation and seismic databases for earthquake generation
	Ding *et al.* (2020) [150]	cGAN	Intensity measures simulation of aftershock seismic events
	Ding *et al.* (2021) [151]	cGAN	Prediction of spectral accelerations of aftershock seismic events
	Matinfar *et al.* (2022) [152]	DCGAN	Generation of spectrum-compatible earthquake accelerograms
	Grijalva *et al.* (2021) [153]	ESeismic-GAN	Generate synthetic magnitude responses of volcano origin event signals
	Florez *et al.* (2022) [154]	WGAN	Data-driven synthesis of broadband earthquake ground motions
	Esfahani et al. (2022) [155]	TF-cGAN	Nonstationary ground-motion simulation in the time-frequency domain
Applications	Fan *et al.* (2021) [156]	SegGAN	Data-driven structural dynamic response reconstruction under seismic load
	Yamada *et al.* (2021) [157]	GAN	Automatic seismic damage investigation for timber houses
	Liao *et al.* (2021) [158], Lu *et al.* (2022) [159]	pix2pix GAN	Automated structural design of shear wall in high rise residential buildings
	Kuurková *et al.* (2018) [160], Ueda *et al.* (2018) [161]	WGAN-GP + VAE	Data correction and seismic structural design of frame building
	Tilon *et al.* (2020) [162]	ADGAN	Anomaly detection GAN for buildings post-disaster analysis
	Zhang *et al.* (2022a) [163]	GAN	Image generation for post-disaster victim detection

Table 2.2: Applications of GANs to seismic-related studies. Source [19]

1D or 2D convolution. This latter technique is discussed in section 2.4. A non-exhaustive list of generative architectures can be found in Table 2.2, reported from [19].

Advantages of generative adversarial learning

GAN offers noteworthy advancements compared to previous methods, such as Variational Autoencoders. The subsequent paragraphs will outline several of these improvements.

(i) **High-Quality Data Generation**. By capturing the features of each sample more accurately, GANs have helped to tackle the blurriness problem found in VAEs. This is due to the discriminator's function, which prioritizes telling apart authentic samples from synthetic ones. Furthermore, only through the discriminator's evaluations does the generator update itself. This forces it to get better at making counterfeit data instances. Importantly, this iterative process hasn't significantly reduced reconstruction quality while generating.

(ii) **Capturing Complex Data Distributions**. The generator is updated via the discriminator. Since the discriminator's task is to look for details that differentiate the synthetic data from those in the given database, the discriminator forces the generator to apprehend more complexity. This type of adversarial training has the particularity of forcing the generator to produce a parametric distribution close to the real distribution of the data.

(iii) **Transfer Learning and Domain Adaptation**. Extending the learning from a previous sub-domain of data readily applies to generative adversarial learning. Transfer learning from a pre-trained discriminator/generator pair can expedite the acquisition of knowledge for a sub-data set. Due to the method's capability to learn complex patterns that can be parametrized, the potential of generalization can be harnessed across different fields without being confined to a single mode. Instead of training the entire network's weights, attention is directed solely toward high-level features, while lower-level features remain frozen during forward computation (see the example of SeismoGen [81]). It has been showcased that:

  - The transfer learning strategy can effectively improve the performance of models trained on synthetic and real samples.
  - It is worth mentioning that transfer learning may be most effective when some relations (low-level features, weights, etc.) exist between source and target domains.

  Finally, because a lower amount of weight is used for the transfer, such an approach can reduce computational costs.

(iv) **Data Augmentation and Synthesis**. Pre-trained GAN can also be used to augment a database. This technique consists of adding the generator outputs from a random distribution to an existing database on which the generator has been trained beforehand. This technique helps with classification when data is either missing or too expensive to collect. GAN can, therefore, be used to improve the accuracy of supervised training.

Limitation of Generative Adversarial Networks

Although GANs showed generation and latent structuring capabilities, they are not without limitations. This section will delve into these constraints.

(i) **Training Instability**

One of the greatest limitations of GAN is its instability during training. Nash equilibrium can be difficult to achieve because when the discriminator or generator capacities are too different, this equilibrium becomes difficult to guarantee (see Section 2.3.3). In these conditions, the loss opponent moves away from the optimal value. The fact of having optimal values for the loss does not justify that the network has reached the optimal values. In addition, the gradient may be unstable, making it impossible to stay at the local minimum despite optimization techniques. In this case, the gradient may be too large or too small (*vanishing gradient*). Overfitting and underfitting are not inevitable, either. What can be observed is a wide disparity between the quality of the input generation and the test generation. Finally, analyzing a GAN requires experience in observing all the intermediate values for rectification, which can become time-consuming. These difficulties highlight the fact that adversarial training is difficult to balance. Research offers various recipes for limiting the network's capabilities, but this remains an open area of research, as mentioned by Roth *et al.*, 2017 [164], Lee *et al.* [165] or Rangwani *et al.*, 2022 [166].

(ii) **Mode collapse**. As a Variational Auto Encoder (VAE), mode collapse is produced when the generator reproduces the same synthetic data independently from the random noise passed through. This problem arises when the generator discovers and perpetuates an example capable of fooling the discriminator, which shows the limits of the latter's ability to evaluate the multitude of data generated effectively. In this case, the discriminator network only focuses on one region of the data. [167]. To overcome this issue, we need to either adapt the data set or introduce noise at the input of the discriminators. The solution involves either reformulating the architecture of the generator and discriminator or adding other normalization and regularisation techniques to overcome the limits.

(iii) **Hyper-parameter sensitivity**. An appropriate hyperparameter tuning has the potential to expedite the training procedure and yield favorable outcomes. However, striking the balanced choice between the Discriminator and the Generator represents a complex undertaking. The GAN is notably responsive to adjustments in multiple hyperparameters, including the learning rate, the scheduler (Linear, cosine, cycling, etc.), the optimization method (Adam [168], Rmsprop [169], Adagrad [170], etc.), the ratio of training between Discriminator and Generator (1:1, 5:1 or 3:1) as well as the batch size. Each hyperparameter wields substantial influence over the training's quality or dynamic, Heusel *et al.*, 2018 [171]. Initiating GAN training anew on a distinct data set, even with a model optimized for a particular task, introduces the necessity of adapting these hyperparameters. Failing to tailor these parameters to the novel context can obstruct the attainment of the Nash equilibrium, a state of balance central to GAN performance. Unexpected model behaviors, divergence, and vanishing or exploding gradient predicaments can ensue if the hyperparameters are not calibrated beforehand. Without relevant expertise, attempting to rectify each of these aberrations in isolation is decidedly ill-advised.

Summary

Generative Adversarial Network (GAN). The generative adversarial learning encompasses two different models: a Generator, G, and a Discriminator, D. These agents play on the min-max game. This is the core of the GAN training. Those two models compete simultaneously: while the Discriminator makes a binary classification between real and fake samples, the Generator tries to mimic the fake samples as closely as possible to the data distribution. A loss function, hence, is used as follows:

$$\min_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$
$$\min_{G} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(D(G(\boldsymbol{z})))]$$

This research has pushed the boundaries of classification tasks for general seismic data and time series. Though powerful, the process could be more manageable as its min-max equilibrium is difficult to ensure. As a result, it can lead to various training problems like mode collapse or dropping and imbalance training. Additionally, hyperparameters must be tuned carefully, or we risk losing all progress.

2.3.4 Adversarial Learning Inference

Adversarial Learning inference (ALI) is considered a bi-directional GAN and represents a significant breakthrough within generative models. The proposed methodology diverges markedly

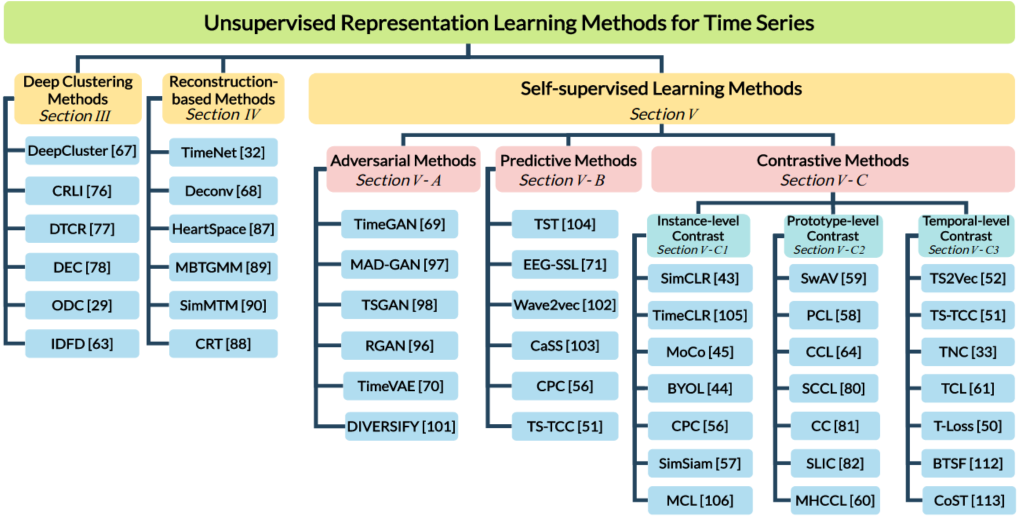


Figure 2.28: Diagram that summarizes existing methods that exist in machine learning for unsupervised tasks applied to time-series, according to Qianwen Meng *et al.*, 2023 [114]. This taxonomy includes deep Clustering Methods, Reconstruction-based Methods, and Self-supervised Learning Methods. Self-supervised learning methods can be further divided into adversarial, predictive, and contrastive methods, depending on the type of pretext tasks employed for acquiring self-supervised signals.

from established techniques such as VAEs, GANs, and AE. ALI introduces an innovative method that intricately incorporates the joint distribution of data and latent space into an analytical framework (see Equation 2.28). Specifically, for the encoder, the formulation takes on the following structure: $q(\boldsymbol{x}, \boldsymbol{z}) = q(\boldsymbol{x})q(\boldsymbol{z} \mid \boldsymbol{x})$. Analogously, for the decoder, the expression assumes the form: $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x} \mid \boldsymbol{z})$. This fusion is meticulously orchestrated through a discriminator, which conducts expert assessment to converge toward the overarching goal of harmonizing distributions. At the core of this approach lies the establishment of an appropriate linkage between observed data and latent space, seamlessly realized through a coherent ensemble of encoders and decoders [172, 112, 173]. Different variants of the same method exist, notably ALICE, Bi-GAN, and Big-BiGAN. Each theme gives some improvements and disadvantages but is not necessarily a different method. In detail, the goal of ALI is to match the joint distribution according to the following mini-max problem:

$$
\begin{aligned}
\min_G \max_D V(D, G) &= \mathbb{E}_{q(\boldsymbol{x})}\left[\log\left(D\left(\boldsymbol{x}, G_z(\boldsymbol{x})\right)\right)\right] + \mathbb{E}_{p(\boldsymbol{z})}\left[\log\left(1 - D\left(G_x(\boldsymbol{z}), \boldsymbol{z}\right)\right)\right] \\
&= \iint q(\boldsymbol{x})q(\boldsymbol{z} \mid \boldsymbol{x}) \log(D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z} \\
&+ \iint p(\boldsymbol{z})p(\boldsymbol{x} \mid \boldsymbol{z}) \log(1 - D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z}
\end{aligned}
\tag{2.28}
$$

The objective is to match joint probability distributions in a manner that $p(\mathbf{x}, \mathbf{z})$ and $q(\mathbf{x}, \mathbf{z})$. Like GAN, When the Generator (encoder or decoder) is too powerful for the Discriminator and vice versa, it becomes difficult to project data in the latent space. Consequently, the generated samples will not follow the joint distributions. This issue is explained by Mao et al., 2017 [174].

The ALI method also verifies the criterion of the *optimal discriminator*, according to the following expression:

$$D^*(\mathbf{x}, \mathbf{z}) = \frac{q(\mathbf{x}, \mathbf{z})}{q(\mathbf{x}, \mathbf{z}) + p(\mathbf{x}, \mathbf{z})} \tag{2.29}$$

Under an *optimal discriminator*, the generator minimizes the Jensen-Shannon divergence between the two joint distributions. The generator reaches its minimum when $p(x, z) = q(x, z)$. If the encoder G_x and the decoder G_z are deterministic then $G_x = G_z^{-1}$ and then vice versa. If we could match the joint distribution together, nothing ensure that this strategy is bijective, or in other words, whether cycling is seriously guaranteed.

$$\mathbf{x} \xrightarrow[G_z]{} \mathbf{z} \xrightarrow[G_x]{} \hat{\mathbf{x}} \tag{2.30}$$

Chunyuan Li *et al.* in their 2017 work "ALICE: Towards Principled Methods for Training Generative Adversarial Networks" [175] proposed a technique that aids the network training through the utilization of cross-entropy (CE) loss. This form of loss can be either explicit or implicit. An Euclidean loss is regarded as explicit; when the distribution alignment is enforced through a discriminator, it becomes implicit. In the research paper titled "Large Scale Adversarial Representation Learning," authored by Donahue *et al.* (2019)[112], a similar approach is pursued. The primary distinction lies in the fact that the Discriminator shares the weight of its various components, and a hinge loss is employed instead of cross-entropy or Earth motion loss. Nonetheless, the ultimate objective remains compatible. View Figure 2.29.

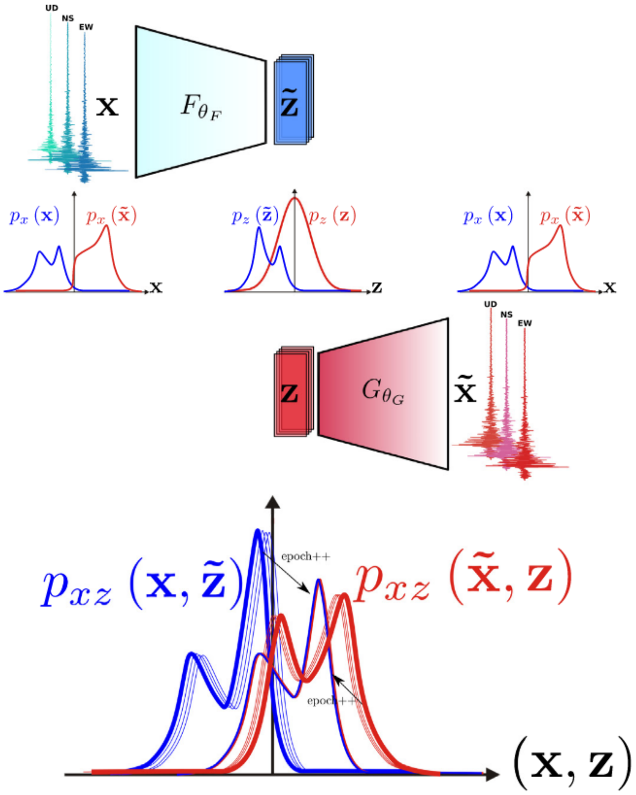


Figure 2.29: Scheme of adversarial learning strategy for joint distributions over. Source Gatti et Clouteau, 2020 [21]

Algorithm 6 The Adversarial Learning Inference algorithm

```
Initialization of network parameters $\theta_d$, $\theta_g$
for $e$ in number of epochs do
    for $k$ in number of discriminator training step do
        Sample mini-batch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from prior $p_g(\boldsymbol{z})$
        Sample mini-batch of $m$ examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data $p_{\text{data}}(\boldsymbol{x})$
        Update the discriminator by ascending its stochastic gradient:
        $\eta_d \leftarrow \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(D\left(\boldsymbol{x}^{(i)}, G_z(\boldsymbol{x}^{(i)})\right)\right) + \log\left(1 - D\left(G_x(\boldsymbol{z}^{(i)}), \boldsymbol{z}^{(i)}\right)\right)\right]$
        $\theta_d \leftarrow \theta_d - \eta_d$
    end for
    Sample mini-batch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from prior $p_g(\boldsymbol{z})$
    Update the generator by ascending its stochastic gradient:
    $\eta_g \leftarrow \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(D\left(G_x(\boldsymbol{z}^{(i)}), \boldsymbol{z}^{(i)}\right)\right) + \log\left(1 - D\left(\boldsymbol{x}^{(i)}, G_z(\boldsymbol{x}^{(i)})\right)\right)\right]$
    $\eta_g \leftarrow \eta_g + \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{x}^{(i)} - G_x(F_x(\boldsymbol{x}^{(i)}))\|$
    $\theta_g \leftarrow \theta_g - \eta_g$
end for
The Adam algorithm is commonly used for that task
```

Application of Adversarial Learning Inference and variants

(i) **Joint Learning and Generation**. The Adversarially Learning Inference (ALI) maps a pair of distributions using the joint distribution. The pair to match are $(\boldsymbol{x}, G_z(\boldsymbol{x}))$ and $(G_x(\boldsymbol{z}), \boldsymbol{z})$. The ALI strategy aims to understand how to project the original data $\boldsymbol{x}$ into the latent space domain, represented by $G_z(\boldsymbol{x})$. Furthermore, reversely, it becomes possible to map random noise $\boldsymbol{z}$ to a sample in the data set using $G_x(\boldsymbol{z})$. The utilization of a joint distribution allows the model to learn complex data distributions. Diverse datasets that might inherently share similarities, such as common features like edges or shoes, could be connected. This method could find a path to domain translation. Further discussion on domain translation will be presented in Chapter 5.

(ii) **Representation Learning** Feature learning or Representation learning is a class of learning techniques that automatically discover hidden patterns in raw data, replacing manual feature extraction. In other words, representation learning aims to discover such hidden features (regularity, trends, correlations, etc.) by passing through abstract representations of such data, often encoded into a lower dimensional space, without crafting specific algorithms. It can be performed in unsupervised, semi-supervised, and supervised way [114]. Supervised learning has succeeded in directly learning semantic information from many labeled samples in recent years. However, this requires a vast amount of annotated training data and is often tailored to specific tasks, making the trained model less transferable. Self-supervised approaches leverage the pretext tasks to autonomously generate labels by utilizing intrinsic information derived from the unlabeled data. Self-supervised learning has made significant leaps fueled by advancements in contrastive learning.

In a nutshell, without a handcrafted representation of data, we let the models learn how different data are linked. Such feature learning is an abstraction process that captures and gathers similarities in data and isolated groups of data that are not an outlier or not representative of the data set. A versatile process could be used as a cross-domain adaptation and for contrastive learning.

(iii) **Enhanced Control over Generation**

Recent investigations in the latent space have shown how controlling generation is important. By appropriately manipulating the latent space, we can perform Fine-Grained Manipulation [176, 177], Customization, Personalization, Data Augmentation [178], Creative Expression, Semantic Control, and achieve Interpretable Outputs. A considerable amount of literature has been published on observing the algebraic aspects that can be employed in the latent space to control the generation aspect of trained generators with an adversarial strategy. The ability to manipulate the latent space has been seriously studied. In "Alan Shoshan's GAN Control" [179, 180], these methods are categorized into two groups.

The first type of method permits relative control. The main idea is to exploit the inherent disentanglement of the latent space. The common and intuitive approach uses Principal Component Analysis (PCA) directions corresponding to the data attributes. Another way to explore the latent space is through off-the-shelf classifiers to extract and predict different attributes (classes and labels for each sample) according to Fard *et al.*, 2023 [181]. If we define the Ψ the off-the-self-classifiers, and S the sought attribute we could define $\Psi(\mathbf{z}, S) \rightarrow \mathbf{z}_i^{fb}, \mathbf{z}_i^{id}$. The $\mathbf{z}_i^{ib}$ is the attribute editing part, and the $\mathbf{z}_i^{fb}$ is the agnostic part, on which we do not try to provide a sense as latent values. The manipulated latent space will control the output of the Generator G, based on a class S. This method is designed to scrutinize and to observe decision boundaries in the latent space, where each side of the boundary corresponds to an opposite semantic attribute. Additionally, we can traverse the latent space and observe the effect of vectors closer and farther from a boundary to increase or decrease the intensity of attributes.

The second type of method provides explicit control over the generation task. The first one is the conditioned version of the GAN, where, through discrete variables, one can enforce control over generation but may not simultaneously assist in controlling multiple attributes. Techniques like StyleRig [182], DiscoFaceGAN [183] for continuous variables offer solutions for translating controls of 3D face rendering models into the GAN-generation process. Other authors propose training GAN [80] with explicit properties and adding various upper parameters during training to create an interpretable explicit control sub-space.

Researchers are developing novel GAN architectures that are more stable, produce diverse samples, and allow for better control. StyleGAN2 [184], BigGAN [185] have demonstrated significant improvements.

(iv) **Latent Space Interpolation** The Gaussian nature of the latent space gives it an essential mathematical stability, distinguishing it from autoencoders (AE) and variational autoencoders (VAE). This stability is further enhanced in unidirectional generative adversarial networks (GANs) and reaches its peak in bidirectional GANs (BiGANs). In BiGANs, we capitalize on the Gaussian distribution of the latent vector, where addition or multiplication by a scalar preserves this Gaussian property. As a result, the latent space has the characteristics of a vector space, which guarantees that all values can be interpreted. This Gaussian adherence allows transparent interpolations of the latent space, ensuring that algebraic transformations produce meaningful and consistent results closely aligned with the training data set. This structural approach has significantly contributed to advancing our understanding of the field.

(v) **Cross-Modal Generation and Translation** In machine learning, adversarial learning inference transfers knowledge from one dataset type to another. The Networks learn the features of the data. This learning process involves learning connections between different distributions. This ability is known as cross-modal generation. This strategy is used for text-to-image, for instance. On the strength of matching different distributions, that method is also employed for multi-modal domain translation. This method was first seen in CycleGAN and then in BiCycleGAN to enable one-to-many mapping [186].

ALICE in Seismology

Gatti *et al.*, 2020, [21] introduce the concept of Adversarial Learning Inference with Cross-Entropy in the seismic research field through the network SeismoALICE. The authors wanted to improve physic-based earthquake numerical simulations from the source to the site (referred as to $\mathbf{x}$) to reproduce historical seismic events and predict future ones. Despite the use of high-fidelity numerical software, namely SEM3D (`https://github.com/sem3d/sem`), such simulations are generally accurate in a 0-5 Hz frequency band because of the lack of data related to the underground geological features and on the seismic context of the region at stake (namely, the characteristics of active faults).
The sensitivity of high-fidelity physics-based earthquake simulation engines, such as SEM3D, to geological and seismological characteristics is extremely high. A large uncertainty on such elements can negatively impact the reliability of the predicted ground motion at the surface. Moreover, even if the latter uncertainty on the underground features of the Earth's crust could be reduced by conductive extensive and expensive seismic tomography and inversion studies in the region of interest, the computational burden of achieving accurate solutions in a larger frequency band, *e.g.*, 0-30 Hz, the cost of such accurate SEM3D simulations would require extremely large computational burden, of the order of more than 100000 CPU hours.
However, the need for 0-30 Hz-accurate earthquake synthetic ground motion is essential to adopt them as input ground motion for conceiving critical infrastructures, such as nuclear power plants. As of now, recorded earthquakes are adopted instead (referred as to $\mathbf{y}$ by Gatti *et al.*, 2020, [21]). However, some regions worldwide, such as metropolitan France, do not have large earthquake catalogs and associated seismograms over extended regions and have sufficient resolution (*i.e.*, a recording station every 100 m). Therefore, Gatti *et al.*, 2020, [21]) attempted to transform simulations 0-5 Hz $\mathbf{x}$ into realistic 0-30 Hz, thanks to ALICE. They split the problem into three parts: $\mathcal{P}_x$, to learn the latent features $\mathbf{z}_x$ of $\mathbf{x}$ and $\mathcal{P}_y$, to learn the latent features $\mathbf{z}_y$ of $\mathbf{y}$ and $\mathcal{P}_z$, to bridge physic-based simulations $\mathbf{x}$ and the recorded data $\mathbf{y}$, through their latent representations $\mathbf{z}_x$ and $\mathbf{z}_y$. The latter is associated with $\mathbf{x}$ and $\mathbf{y}$ through ALICE. ALICE is also employed to infer $\mathbf{z}_y$ from $\mathbf{z}_x$, rendering hybrid time-histories. However, this approach is undeniably intricate, demanding a multitude of Discriminators and Generators to execute a singular task: transforming physics-based data into ground-motion data. Additionally, the neural network architecture cannot facilitate data projection into Gaussian space and subsequent retrieval. Consequently, achieving a one-to-many mapping is unfeasible. We develop this in another section 4.1.

Limitation of Adversarially Learning Inference and variants

Limitations. The main limitation of GAN is presented in [187]. Mode collapse presents. Non-convergence. Defining the appropriate Agent (Generator and Discriminator) to achieve the Nash equilibrium is hard.

Application to time-series

Different strategies are proposed for the task:

(i) TimeGAN[188] is primarily applied within the domain of time-series data. As outlined in [114], [188] amalgamates the advantages of auto-regressive models and the GAN framework. This combination is achieved by creating an embedded space that is jointly optimized using supervised and adversarial objectives. Subsequently, the model becomes adept at capturing and replicating the temporal dynamics inherent in the training data during sampling. Notably, [188] assumes an essential role in establishing the mapping between the input features and the corresponding latent representations, facilitating the generation of meaningful time-series data.

(ii) MAD-GAN[137] The Multivariate Anomaly Detection for time-series Data with Generative Adversarial Networks (MAD-GAN) identifies intricate multivariate correlation in the time-series dataset for anomaly detection.

(iii) DIVERSIFY [189] This method employs an adversarial learning strategy to optimize two critical objectives concurrently: firstly, to maximize the *worst-case* distribution scenario, and secondly, to minimize distribution divergence. This entails segmenting time-series data into discrete latent sub-domains, with the specific intention of amplifying the distribution gap at the segment level, thereby preserving the richness of diversity. Simultaneously, DIVERSIFY [189] undertakes the task of diminishing distribution divergence across these discerned latent domains, ultimately yielding domain-invariant representations. This methodology capitalizes on the inherent diversity ingrained within latent distributions inherent to time-series data, which often encompass distinct and nuanced activity patterns exhibited by multiple individuals.

Summary

The Adversarial Learning Inference combines the power of an Auto-encoder and the power of the Generative adversarial Network. Using the uncertainty quantification technique has led to manipulating joint distribution. The solution of joint distribution allows the generation problem to be solved. Also, the bijectivity is conserved through the cross entropy, i.e., one maintains a clear reconstruction consistency. This technique's advantages have showcased a novel way to control many aspects of our dataset. The Loss function used is :

$$\min_G \max_D V(D, G) = \mathbb{E}_{q(\boldsymbol{x})} \left[\log \left(D\left(\boldsymbol{x}, G_z(\boldsymbol{x})\right)\right)\right] + \mathbb{E}_{p(\boldsymbol{z})} \left[\log \left(1 - D\left(G_x(\boldsymbol{z}), \boldsymbol{z}\right)\right)\right]$$

Adversarial Learning Inference offers joint learning and generation, a better representation of latent value, and a clear understanding of generations; this technique is also applied to domain translation. Such techniques find a way of utilization in Seismology through the SeismoALICE, and more widely in time series problems. However, this technique suffers from Mode collapse and convergence issues as with other frameworks. Moreover, the Nash equilibrium technique could take several iterations to be achieved.

2.3.5 Contrastive Learning

Contrastive Learning (CL) can generate augmented views of raw data through diverse transformations and learn representations by contrasting positive samples against negative samples. CL is a powerful deep learning technique that has demonstrated effectiveness in learning shared and meaningful representations from unlabeled, similar data through optimizing self-discrimination tasks: similar data is brought together while dissimilar ones are pushed apart. Figure 2.30 illustrates Contrastive Learning. In unsupervised scenarios, contrastive Learning helps to discern the similarity between data points, enabling the Learning of high-level features about the data even before engaging in specific tasks like classification or segmentation without the need for explicit labels. Contrastive Learning removes the need to reconstruct the entire input.

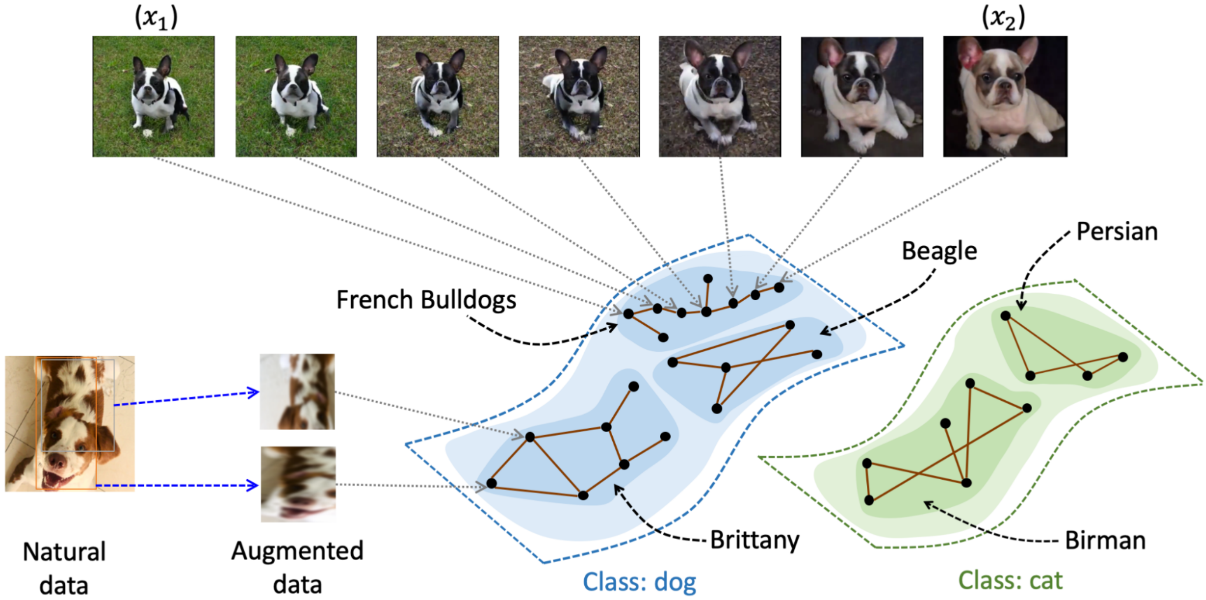


Figure 2.30: Example of representation of contrastive learning from, HaoChen *et al.*, [190]

Different losses exist to maximize agreement between pairs, and each has proven how to maximize the pair of plausible values. For instance, in Chen *et al.*, 2020 [113], the contrastive loss in batch of N values is provided by the following expression:

$$l_{i,j} = -\log\left(\frac{\exp(\mathrm{sim}(\mathbf{z}_i,\mathbf{z}_j)/\tau))}{\sum_{k=1}^{2N} 1_{[k\neq i]}\exp(\mathrm{sim}(\mathbf{z}_i,\mathbf{z}_j)/\tau)}\right) \tag{2.31}$$

The sign minus expresses the fact that the loss should be maximized, whereas the function sim corresponds to the cosine of the angle between two vectors in a Hilbert space, according to the following expression (issued from the scalar product):

$$\mathrm{sim}(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^\top\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

The term τ in Equation 2.31 is generically referred to as to *temperature.* This loss is called NT-Xent, but others exist, such as NT-Logistic, Margin Triplet, and CLIP (Contrastive Language-Image Pre-training, Ford *et al.*, 2021 [**?**]). Algorithm 7 illustrates the contrastive learning training scheme proposed by Chen *et al.*, 2020 [113].

Three distinct caterogories of CL exist: the instance-level, prototype-level and temporal levels representations.

(i) **Instance-level CL models** To aggregate similar information, data augmentation is adopted to transform original data into a new embedding space. Within this embedding space, the

Algorithm 7 Contrastive Learning algorithm by Chen *et al.*, 2020 [113]

```
input:batch size $N$, constant, $\tau$, structure of $F, G, \mathcal{T}$
for sampled mini-batch, $\{\mathbf{x}_{\mathbf{k}}{}_{k=1}^{N}$ do
    for $k \in \{1, \cdots, N\}$ do draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$ $\tilde{\mathbf{x}}_{2k-1} = t(\mathbf{x}_k)$ $\mathbf{h}_{2k-1} = F(\tilde{\mathbf{x}}_{2k-1})$ $\mathbf{z}_{2k-1} = H(\mathbf{h}_{2k-1})$
        $\tilde{\mathbf{x}}_{2k} = t'(\mathbf{x}_k)$ $\mathbf{h}_{2k} = F(\tilde{\mathbf{x}}_{2k})$ $\mathbf{z}_{2k} = H(\mathbf{h}_{2k})$
    end for
    for $i \in \{1, \cdots, 2N\}$ and $j \in \{1, \cdots, 2N\}$ do $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_i\|)$
    end fordefine $l(i, j)$ as :
        $l_{i,j} = -\log \left( \frac{\exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)} \right)$
        $\mathcal{L} = \frac{1}{2N} \sim_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)]$
    Update the Networks $F$ and $G$ to minimize $\mathcal{L}$.
end for
Return $F(\dot{})$ and discard $G(\dot{})$
The Adam algorithm is commonly used for that task
```

augmentation derived from the same sample is treated as a positive pair, while those from different samples are regarded as negative pairs. In training, the models are optimized by maximizing the similarity between the representations of positive pairs while minimizing the similarity between the representations of negative pairs. In this category, one can cite TimeCLR [191], MoCo [192], SimCLR [113], BYOL [193], CPC [194], SimSiam [195], MCL [196], see Table 2.3

Nevertheless, these methods are rather memory-intensive implementations, and maintaining the consistency of their representations becomes difficult when randomly extracting negative samples from an additional storage space.

(ii) Prototype-level CL models

The traditional Contrastive Learning algorithm (presented at the beginning of Section § 2.3.5) actively approaches pairs of similar distributions without needing semantic annotations. The technique mentioned above is focused on transforming the samples and evaluating the abstract representation of data using contrastive loss. However, such a strategy faces limitations in gathering two pairs of semantically identical distributions: a trained model could be considered different through contrastive evaluation.

To overcome a common issue in instance-level contrastive learning, i.e., when similar samples are mistakenly treated as negatives (if a point is used as an anchor, a point similar to it is called positive, and a point dissimilar to it is called negative.), the Prototype Constrative Learning is introduced. In addition to the sample instance, the related annotation is passed through contrastive learning. Therefore, the "*prototypical contrastive learning is referred to as a representative embedding of a group semantically similar instance*" [197]. One of the algorithms utilized in this process is named *ProtoNCE*, which describes better

Models	Main Contributions	Backbones	Datasets	Evaluations
SimCLR	SimCLR advocates for a learnable nonlinear transformation bridging the representation and the contrastive loss to enhance the quality of learned representations, and highlights the significance of larger batch sizes, more training steps, and the composition of data augmentations	ResNet	ImageNet, CIFAR-10	Classification, Transferability
TimeCLR	TimeCLR extends SimCLR to the time series domain, amalgamating the benefits of dynamic time warping data augmentation tailored for univariate time series, and the potent feature extraction capability of InceptionTime, thus facilitating the acquisition of representations.	InceptionTime	Hand Atlas	Classification
MoCo	MoCo maintains a dynamic queue to enrich the set of negative samples, and proposes a slowly progressing key encoder, implemented as a momentum-based moving average of the query encoder, to preserve the consistency of key representations.	ResNet	ImageNet, CIFAR-10	Classification, Transferability
BYOL	BYOL removes the need for using negative samples and employs a predictor on top of the online network to learn the mapping from the online encoder to the target encoder, which helps prevent mode collapse.	ResNet	ImageNet, CIFAR, SUN397, VOC07, DTD	Classification, Transferability
CPC	CPC extracts compact latent representations to encode predictions over future observations by combining autoregressive modeling and noise-contrastive estimation with intuitions from predictive coding.	ResNet, GRU	LibriSpeech, ImageNet, BookCorpus	Classification
SimSiam	SimSiam learns representations by using Siamese architectures without negative sample pairs, large batches, and momentum encoders. SimSiam tackles the mode collapse problem by using the stop-gradient mechanism.	ResNet	ImageNet, VOC07, COCO	Classification, Transferability
MCL	MCL learns representations through the injection of noise. Inspired by label smoothing, MCL adopts mixup that creates new samples by convex combinations of training examples and predicts the strength of the mixing component based on 2 data points and the augmented sample.	FCN	UCR Datasets, UEA Dataset	Classification, Transferability

Table 2.3: Summary of instance-level contrastive methods for time series representation learning. Source [114]

the data distribution. The adopted loss reads:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{n} -\log \frac{\exp(\mathbf{v}^i \cdot \mathbf{v}'^j/\tau)}{\sum_{i=0}^{r} \exp(\mathbf{v}^i \cdot \mathbf{v}'^j/\tau)} \tag{2.32}$$

Formula in with $\mathbf{v}^i$ is the positive sample of instance i, $\mathbf{v}'^j$ is a sample containing one positive embedding, and the r negative sample for other instance i. τ is a temperature hyper-parameter. Those sample coming from feeding $\mathbf{x}^i$ to a momentum encoder: $F(\mathbf{x}^i) = \mathbf{v}^i$. Protype-level contrastive learning models aim to disrupt the independence between samples by bringing contrastive learning and clustering to capture the hierarchical semantic structure of the dataset, implicitly shared by samples. Within this category those methods can be included : SwaV [198], PCL [197], CCL [199], SCCL [200], CC [201], SLIC [202], MHCCL [203]. View Table 2.4

Unfortunately, the traditional approaches have a drawback in requiring previous knowledge to identify the number of clusters in advance. When dealing with unlabeled time-series data, this task becomes non-trivial. Moreover, their flat clustering algorithms are limited to capturing a single semantic structure. This restriction is troublesome since it makes it impossible to depict the intricacies of the data distributions, Zhixiong. *et al.*, 2022 [204].

(iii) **Temporal-level CL models** This strategy sprouted from the limitations of contrastive learning for time series, compared to its accuracy in computer vision tasks. The application of contrastive learning models often ignored the intricate characteristics of such data. While instance-level contrastive learning models can only capture general representations of the entire time series, in temporal-level contrastive learning, these models focus on capturing scale-invariant representations at each time sequence. The research found that mixing both Instance-level and temporal-level representation enhances the capability of contrastive learning methods in capturing complexities inherent in time-series data. In this category, the following works can be included : TS2Vec [205], TS-TCC [206], TNC [207], TCL[208], T-Loss [209], BTSF [210], CoST [211].

Table 2.5 shows a non-exhaustive list of all CL models for time-series analysis.

Summary of Contrastive Learning

Model learning, by contrast, will have a discriminative representation of raw data. Such a technique eliminates the need to reconstruct the entire input to capture similarities inherent in the data. A trained contrast model can help structure unlabeled data and learn complex representations. It can be trained alone for clustering purposes or along generative tasks. This strategy creates clusters for similar data and adopts a loss function that compares pairs of values to maximize their similarity. Among the various existing losses, we can mention NT-Xent, for example, that for two pairs $\mathbf{x}_i, \mathbf{x}_j$ reads:

$$l_{i,j} = -\log \left(\frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)} \right)$$

Constrative Learning encompasses the instance level, the Prototype Level, and the temporal-level representation for times series. Finally, such a technique is used to aggregate samples, improve classification segmentation, and guide the encoder in extracting relevant features common to the same cluster.

Models	Main Contributions	Backbones	Datasets	Evaluations
SwAV	SwAV optimizes the learned representations by developing a swapped prediction mechanism to simultaneously perform scalable online clustering while enforcing the consistency between cluster assignments produced for different augmented views of the same sample.	ResNet	ImageNet, VOC07, COCO	Classification, Transferability
PCL	PCL formulates prototypical contrastive learning as an Expectation-Maximization algorithm to perform clustering and representation learning iteratively. The E-step aims to find the distribution of prototypes via clustering and the Mstep aims to optimize the network via contrastive learning.	ResNet	ImageNet, VOC07, Places205, COCO	Classification, Clustering, Object Detection
CCL	CCL refines the learned representations acquired through deep convolutional neural networks by discovering dataset clusters with high purity and typically few samples per cluster, and leverage these cluster assignments as the potentially noisy supervision.	CNN	BBT-0101, BF0502, ACCIO	Classification, Clustering
SCCL	SCCL leverages contrastive learning for short text clustering to promote better separated and less dispersed clusters. It effectively combines the top-down clustering with the bottom-up instance-wise contrastive learning to achieve better inter-cluster distance and intra-cluster distance.	DistilBERT	AgNews, Tweet, SearchSnippets, StackOverflow, Biomedical, Googlenews	Clustering
CC	CC seamlessly integrates deep clustering and representation learning by revealing that the row and column of the feature matrix intrinsically correspond to the instance and cluster representation when projecting instances into a subspace whose dimensionality is equal to the cluster number.	ResNet	CIFAR-10, CIFAR-100, STL-10, ImageNet-10	Classification, Clustering
SLIC	SLIC combines iterative clustering with multi-view encoding and temporal discrimination to learn view-invariant video representations and fine-grained motion features. The clustering assignments are used to guide the sampling of positive and negative pairs for updating representations.	CNN	UCF101, HMDB51, Kinetics400	Classification, Video Retrieval
MHCCL	MHCCL incorporates the implicit semantic information obtained from hierarchical clustering to guide the construction of contrastive pairs. MHCCL exploits downward masking to filter out fake negatives and supplement positives, while also employing upward masking to refine prototypes.	ResNet	HAR, WISDM, SHAR, Epilepsy, UEA Datasets	Classification

Table 2.4: Summary of prototype-level constrative methods for time series representation. Source [114]

Models	Main Contributions	Backbones	Datasets	Evaluations
TS2Vec	TS2Vec utilizes multi-scale contextual information at both timestamp-level and instance-level to distinguish positive and negative samples, thereby improving the generalization capability of the representation model and effectively handling time series data with missing values.	Dilated CNN	UEA Datasets, ETT Datasets, Electricity	Classification, Forecasting
TS-TCC	TS-TCC constructs simple yet efficient time-series-specific augmented views to perform temporal and contextual contrasting, and designs a tough cross-view prediction task to learn the robust and discriminative representations.	CNN, Transformer	HAR, Sleep-EDF, Epilepsy, FD	Classification, Transferability
TNC	TNC learns the underlying dynamics of non-stationary signals and models the progression over time by defining a temporal neighborhood. It incorporates concepts from Positive-Unlabeled learning to account for potential bias introduced in sampling negative examples for contrastive loss.	Bidirectional RNN	Simulation, ECG Waveform, HAR	Classification, Clustering
TCL	TCL learns representations for time series that allow optimal discrimination of different time segments based on the temporal non-stationary structure captured by nonlinear independent component analysis	Nonlinear ICA	MEG	Classification
T-Loss	T-Loss learns scalable representations by taking highly variable lengths and sparse labeling properties of time series data into account. It employs an efficient triplet loss with time-based negative sampling to differentiate anchors from negatives, and assimilate anchors with positives.	Causal CNN	UEA Datasets, UCI Datasets, UCR Datasets	Classification, Transferability
BTSF	BTSF applies dropout to generate diverse views for representation learning, and devises iterative bilinear temporalspectral fusion to explicitly model pairwise cross-domain dependencies for discriminating and enriching representations in a fusion-and-squeeze manner.	Causal CNN	HAR, Sleep-EDF, ECG Waveform, ETT Datasets, Weather, SAaT, WADI, SMD, SMAP, MSL	Classification, Forecasting, Anomaly Detection
CoST	CoST simulates interventions on the error variable via data augmentation and exploits prior knowledge to learn time series representations. It leverages inductive biases in the model architecture to learn disentangled seasonal and trend representations via contrastive learning.	Causal CNN	ETT Datasets, Electricity, Weather	Forecasting

Table 2.5: Summary of temporal-level contrastive methods. Source [114]

2.4 Neural network architectures for time-series

A neural network architecture is a nonlinear computational method inspired by brain cells that includes activation functions, weights, and connection to the author neuron. This architecture does not capture relevant features using a handcrafted technique or code. Unlike conventional algorithms, such an infrastructure could be seen as many computing units that tune themselves based on the response of the outsides; the collection of neurons learns through examples and is corrected by the research of satisfying a loss function. The update is made using the back propagation technique, which distributes the correction values through an operation graph. A trained architecture could be employed for new data that has not been seen before. Of course, if the sample test does not represent the trained dataset distribution, this latter fails to succeed. A neuron is connected to the stack of previous neurons, according to the following expression [212]:

$$\mathbf{x}_k^l = \mathbf{b}_k^l + \sum_{i=1}^{N_{l-1}} \mathbf{w}_{ik}^{l-1} \mathbf{y}_i^{l-1} \tag{2.33}$$

Formula in which :

- $\mathbf{y}_i^{l-1} = f(\mathbf{x}_i^{l-1})$, f is an activation functions (ReLU, LeakyReLU, Tanh, etc ...)
- the subscript k here represent the k^{th} neuron
- the superscript l refers to the l^{th} layer
- $\mathbf{x}_k^l$ is defined as the input
- $\mathbf{w}_{ik}^{l-1}$ is the kernel from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l

We will first introduce and explain the Convolutions Neural Network before tackling the Self-Attention technique, which is based on a more robust architecture for analyzing vast and complex data types. In our case, it is the temporal earthquake data. Classical architecture using 1D CNN struggles in capturing long-range dependencies despite trying to increase the length of the kernel, use dilation, and zero padding [213]. This is the reason behind the research of proper architecture to fill the gap. The first to achieve an attractive result was the Transforme. However, this lather was designed for word translation by exploring cross-correlation and embedding positions to reach out more appropriate information from text sequences. Even though the different application cases have been demonstrated, this architecture could be adapted for images. We will have to wait for the Conformer, tailored for time series, to outperform the previous above.
The Conformer surpasses by 15% the Transformer, [121] in different state-of-art architecture in Denoising, Dereverberation, and Super-Resolutions. On the strength of these advantages, we will use the conformer component in our encoder and decoder architecture to achieve, first, a better quality of reconstruction for waveform and, second time, to improve the quality of the generation, as CMGAN in Abdulatif *et al.*, 2022 [119], and to get rid of mode collapse and mode dropping issues. The complexity of such architecture helps surpass all the different issues that we have faced before.
The appropriate architecture is a must, but with the appropriate cost function to train our architecture, this improvement will be useful. Time series signals need to be explored in the time domain, and further exploration should be done in the frequency domain. More explanation is provided in the subsection 2.5. The main architectures adopted in this work are described in the following sections.

2.4.1 Convolution Neural Network

There is a historical reason behind the utilization of the 1D convolution layer. The first time convolution layers were utilized was for images and videos and were dedicated to classification tasks. In this sense, the operation was performed by manipulating 2D and 3D data. 1D Convolutional layers, instead, were a recent invention to deal with 1D signals. The main reason behind that task is the computational task. To proceed with $N \times N$ images, we will need to use $K \times K$ kernel, which has $O(N^2K^2)$ complexity compared to the 1D signal, which has a complexity of $O(NK)$, Kiranyaz *et al.*, 2015 [214]. The computational cost is lower. Rather than utilizing a 2D Convolutional method with a spectrogram generated from the 1D signal, we can straightforwardly opt for a 1D Convolutional Neural Network approach to efficiently capture the pattern structure of seismic time series directly from the waveform data. The 1D CNN excels in feature extraction and pattern recognition, time series data, and also for natural languages. The convolution operation employs a discrete learnable kernel that can be stridden and dilated. The kernel is the learnable weight matrix that discretizes the convolutional operator (discrete kernel). After each convolution, the stride is the lap or the number of steps to move the kernel. The dilation corresponds to the space between elements within the kernel, affecting the receptive field without changing the number of parameters. As default, dilation equals one, and a dilation of 2 or more skips one or more consecutive pixels (or time-steps) to be multiplied by the kernel weights. Several convolution layers are usually stacked upon each other. In the case of 1D time series, the formulation of the discrete convolution reads:

$$\mathbf{x}_k^l = \mathbf{b}_k^l + \sum_{i=1}^{N_{l-1}} \text{conv1D}(\mathbf{w}_{ik}^{l-1}, \mathbf{s}_i^{l-1}) \tag{2.34}$$

The convolution is performed by $\text{conv1D}(\mathbf{w}_{ik}^{l-1}, \mathbf{s}_i^{l-1}) = \mathbf{w}_{ik}^{l-1} \star \mathbf{s}_i^{l-1}$. The $\star$ symbol indicates the cross-correlation operation[3], usually, this operation is followed by normalization and, finally, an activation function $h(.)$ (Tanh, ReLU, LeakyReLU, for example).

- the subscript k here represent the k^{th} neuron
- the superscript l refers to the l^{th} layer
- $\mathbf{x}_k^l$ is defined as the input
- $\mathbf{s}_i^{l-1}$ is the output of the i^{th} neuron at layer $l-1$
- $\mathbf{w}_{ik}^{l-1}$ is the kernel from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l

See the Forward CNN in the Figure 2.31. However, this type of architecture quickly showed limitations in capturing the long-term dependencies inherent in acceleration data. As a result, several variants of this architecture have been proposed to address this challenge. Gated convolution or Gated CNN is introduced in seismology by [140]. The Gate represents a mask applied in the first layers of the decoder after the reshaping of the linear projection of the values from the latent space. This is developed to better capture the sequential and hierarchical structure. It is composed of CNN block and GLU. As we will see after, the Gated CNN could be seen as the

---

[3] The cross correlation between both function f and g is defined as :

$$f \star g)(\tau) \triangleq \int_{-\infty}^{\infty} \overline{f(t)} g(t+\tau)\, dt$$

τ is the stride also called the *lag*

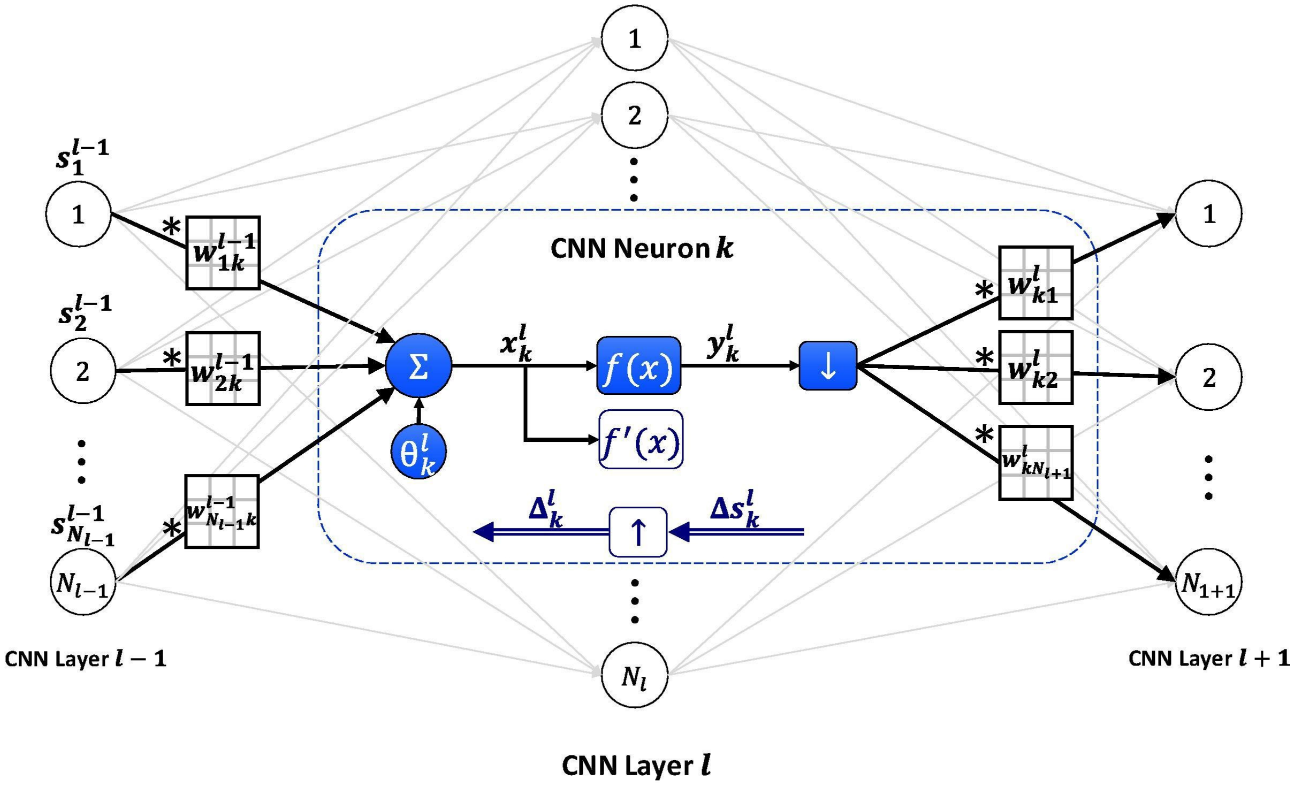


Figure 2.31: CNN connection layer, in Forward process. ©Kiranyaz *et al.*, 2015 [214]

precursor of the Attention technique (see §2.4.3). In the process, The trained neural network should identify patterns and pay different attention through the mask. The mask is represented by applying a sigmoid and a product in the sense that Hadamard (i.e., element-wise product) forces the network to view long-range dependencies because, in time series data, the correlation is seldom local. Pixelshuffle is used for upsampling and effectively solves "checkerboard artifacts". Super-resolved images first revealed such checkerboard patterns. This phenomenon, according to Kinoshita *et al.*, 2020 [215], introduces a distortion in the output images, and it is essentially caused by using upsampling blocks in linear multi-rate systems. The checkerboard is caused by forward-propagation in the upsampler when using a transposed Convolution layer, for example, and by backward-propagation with a stridden convolutional layer. The PixelShuffle rearranges the elements in a tensor according to an upscaling factor. A signal of resolution $H \times W \times C$ is transformed $rH \times rW \times C$, r is the factor of upscaling, according to the following expression :

$$\begin{aligned} f^1(\mathbf{x}^{LR}; W_1, b_1) &= \phi(W_1 * \mathbf{x}^{LR} + b_1) \\ f^l(\mathbf{x}^{LR}; W_{1:l}, b_{1:l}) &= \phi(W_1 * f^{l-1}(\mathbf{x}^{LR}) + b_l) \end{aligned} \tag{2.35}$$

where $l \in (1, L-1)$ represent the layer index and $\mathbf{W}_l$, b_l are learnable weights and biases respectively. $\mathbf{W}_l$ represents the $n_l \times k_l$ "convolution tensor", including n_l features of layer l. $n_0 = C$ is the number of channels. The value k_l represents the size of the discrete filter at the layer l. b_l represents the bias at the same layer. ϕ is the non-linear activation function. The last layer f^L converts the input features $\mathbf{x}^{LR}$ into a super resolution output, $\mathbf{x}^{SR}$ [216]. This strategy was proposed to generate data to improve classification, generation, and reconstruction tasks.

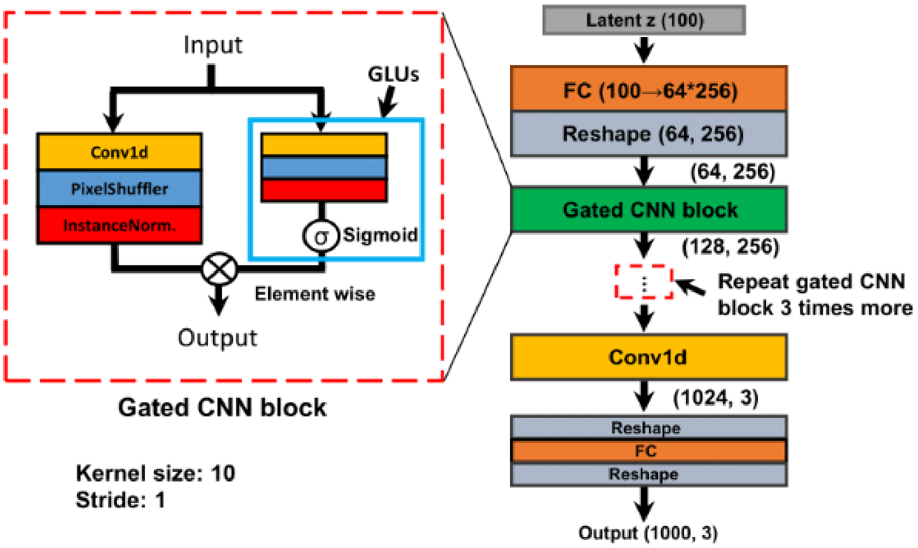


Figure 2.32: Gated Convolutional Neural Network. Courtesy of [140]

2.4.2 Linear Layer

Linear or fully-connected or dense layers connect every input neuron to every output neuron and are commonly used in neural networks. The input $\mathbf{X}$ and the output $\mathbf{H}$ are considered as vectors of respectively dimension N and n, and the weight $\mathbf{W}$ is a $n \times N$ 2D vector. A linear layer is designed as follows:

$$\mathbf{x}_l = \mathbf{x}_{l-1}\mathbf{W}^\top + \mathbf{b}_{l-1} \tag{2.36}$$

Formula in which we are the neurons $\mathbf{W} \in \mathbb{R}^{d_N \times d_n}$, the input $\mathbf{x}_{l-1} \in \mathbb{R}^{d_n}$, the output $\mathbf{x}_l \in \mathbb{R}^{d_N}$ and the bias $\mathbf{b}_{l-1} \in \mathbb{R}^{d_N}$

2.4.3 Scaled Dot-Product and Self-Attention Mechanism

When dealing with audio-type data (i.e., time series such as seismic accelerations), capturing long dependencies is a difficult task due to the limitations of the convolutional kernel used in many models [217], [218]. This fixed kernel length implies a restricted local receptive field, meaning it can only capture information from a small input neighborhood at a time. As a result, as the time series passes through multiple convolutional layers, the importance given to long-range dependencies diminishes. This becomes problematic, especially for smaller models, as they may not have enough capacity to capture the complex relationships in the data. Contrary to what one might think, increasing the kernel size does not help capture these dependencies, but it increases the computational burden and reduces the network efficiency [213].

The optimization algorithm that trains the model struggles to effectively coordinate multiple layers to capture and represent these long dependencies. For this reason, Self-Attention mechanisms (or Intra-Attention) are widely adopted in computer vision to better capture dependencies within the data at different distant positions. This challenge has been acknowledged and explained by Vaswani *et al.*[219] in their seminal paper on Transformer architecture, which proposes an alternative approach to addressing long-range dependencies in models, called *Scaled Dot-Product Attention*. The Scaled Dot-Product Attention proposed by Vaswani *et al.*[219] renders an *attention map* that highlights hidden intra-sample correlations and dependencies, among different parts of it (such as the correlation between different time steps within a time-series or between different words in a phrase). Self-Attention has been developed in the framework of language models, where data is generically represented by a sequence of words, represented by $(\mathbf{x}_n)_{1 \leq n \leq N_l} \subset \mathbb{R}^{d_X}$. Alternatively, the input can be interpreted as a collection of N_l feature maps, $\mathbf{x} \subset \mathbb{R}^{N_l}\mathbb{R}^{d_X}$, as proposed in their work on Self-Attention GAN (SAGAN) by Zhang *et al.*, 2019 [220], who applied similar attention mechanism as the one proposed by Vaswani *et al.*, 2017 [219] (and described below) to convolutional maps in deep CNN, so to capture relevant information in convolution feature maps.

In their seminal work on Transformers, Vaswani *et al.*, 2017 [219] adopted the language model framework, where each word of a sentence $(\mathbf{x}_n)_{1 \leq n \leq N_l}$ or N_l feature maps $\mathbf{x}$ can be embedded into learnable keys-values pairs, referred as to $(\mathbb{K}, \mathbb{V})$, with $\mathcal{K} : \mathbf{x} \in \mathbb{R}^{N_l} \times \mathbb{R}^{d_X} \mapsto \mathbf{W}_K.\mathbf{x} = \mathbb{K} \in \mathbb{R}^{d_K} \times \mathbb{R}^{d_X}$ and $\mathcal{V} : \mathbf{x} \in \mathbb{R}^{N_l} \times \mathbb{R}^{d_X} \mapsto \mathbf{W}_V.\mathbf{x} = \mathbb{V} \in \mathbb{R}^{d_V} \times \mathbb{R}^{d_X}$, with $\mathbf{W}_K \in \mathcal{M}_{d_K,N_l}(\mathbb{R})$, $\mathbf{W}_V \in \mathcal{M}_{d_V,N_l}(\mathbb{R})$. The Scaled Dot-Product Attention proposed by Vaswani *et al.*, 2017 [219] draws (self-)attention maps by firstly evaluating the "projection" $\mathbb{C}_{ij} = \mathbb{Q}_i\mathbb{K}_j^T \in \mathcal{M}_{d_Q,d_K}(\mathbb{R})$ of the data key $\mathcal{K}(\mathbf{x}_j)$ onto the list of d_Q queries obtained by linear embedding $\mathbb{Q}_i = \mathbf{W}_Q.\mathbf{x}_i = \mathcal{Q}(\mathbf{x}_i) \in \mathbb{R}^{d_Q}$ and $\mathbf{W}_Q \in \mathcal{M}_{d_Q,d_{N_l}}(\mathbb{R})$. The output of the projection expresses some sort of "template match" between them. As shown in Figure 2.33, an attention map is obtained by capturing relevant features, obtained by: (i) a scaling of $\mathbb{C}_{ij}$ by a factor $\frac{1}{\sqrt{d_K}}$ and (ii) an optional learnable mask $\mathbf{M}$, multiplied element-wise by $\tilde{\mathbb{C}}_{ij} = \frac{1}{\sqrt{d_K}} \cdot \mathbb{C}_{ij}$, so to obtain $\mathbb{S}_{ij} = \frac{1}{\sqrt{d_K}}\mathbf{M} \odot \mathbb{C}_{ij}$ (adopting the Hadamard product, indicated by $\odot$). Once $\mathbb{S}_{ij}$ assembled, a column-wise softmax is applied to each l^{th} column $\mathbf{s}_{ij}^{(l)} = [\mathbb{S}_{ij}]_{.l}$, acting as a normalization, so to obtain $\boldsymbol{\beta}_{ij}$, whose entries read:

$$\left[\boldsymbol{\beta}_{ij}\right]_{kl} = \frac{\exp([\mathbb{S}_{ij}]_{kl})}{\sum_{m=1}^{N} \exp([\mathbb{S}_{ij}]_{ml})}, \quad \boldsymbol{\beta}_{ij}^{(l)} = \text{softmax}\left(\mathbf{s}_{ij}^{(l)}\right) \tag{2.37}$$

As shown in Figure 2.33, the attention map $\boldsymbol{\beta}_{ij}$ is then adopted to weight the embedded value $\mathbb{V}_i = \mathcal{V}(\mathbf{x}_i)$, so to consider its correlation with input $\mathbf{x}_j$, *i.e.*:

$$\text{Attention}(\mathbb{Q}_j, \mathbb{K}_j, \mathbb{V}_i) = \mathbb{H}_{ij}\mathbb{V}_i \tag{2.38}$$

Where $\mathbb{H}_{ij} = \mathcal{H}(\boldsymbol{\beta}_{ij})$ is an optional linear layer, defined as:

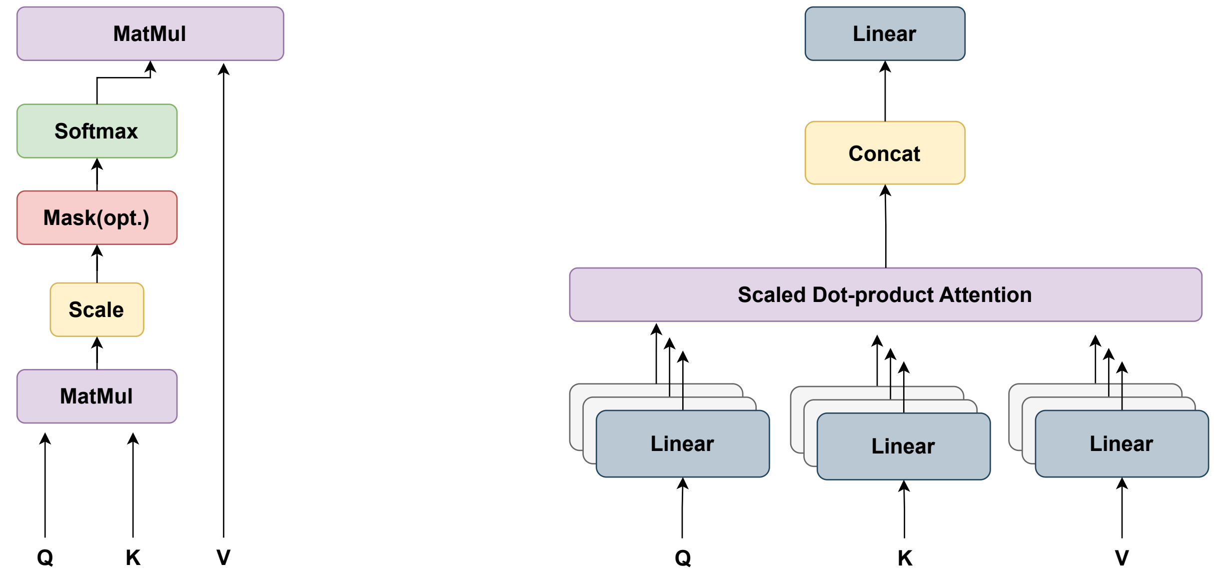
$$\mathcal{H} : \boldsymbol{\beta} \in \mathbb{R}^{d_Q} \times \mathbb{R}^{d_K} \mapsto \mathbf{W}_H.\boldsymbol{\beta} = \mathbb{H} \in \mathbb{R}^{d_H}$$

Where the $\mathbf{W}_H \in \mathbb{R}^{d_H} \times \mathbb{R}^{d_Q}$. In practice, $d_K = d_Q = d_H = d_V$. The idea behind the Scaled-Dot-Product Attention proposed by Vaswani *et al.*, 2017 [219] (see Equation 2.38) was retrieved and slightly modified by Zhang *et al.*, 2019 [220], so be adapted to convolutional maps in the so-called self-attention GAN. In their work, Zhang *et al.*, 2019 [220] referred $(\mathbf{x}_n)_{1 \leq n \leq N_t}$ as a sequence of convolutional maps outputted by a hidden layer of a CNN (see CNN description in Section 2.4.1). The key-value-query embedding is performed through an element-wise convolution (*i.e.*, a convolution with kernel of size 1) and a subsequent MaxPooling. Zhang *et al.*, 2019 [220] neither mention the learnable mask $\mathbf{M}$ nor apply the scaling factor $\frac{1}{\sqrt{d_K}}$, so that $\mathbb{C}_{ij} = \mathbb{S}_{ij}$. Moreover, Zhang *et al.*, 2019 [220] recombined the attention output sequence $\mathbf{o} \subset \mathbb{R}^{N_l}\mathbb{R}^{d_X}$, whose i^{th} occurrence reads:

$$\mathbf{o}_i = \sum_{h=1}^{d_X} \boldsymbol{\beta}_{hi}.\mathbb{V}_h \tag{2.39}$$

with a residual connection to $\mathbf{x}_i$, according to the expression (see Figure 2.34):

$$\mathbf{y}_i = \gamma\mathbf{o}_i + \mathbf{x}_i \tag{2.40}$$



(a) Scaled Dot Product Attention Architecture. Q, K, and V are the Queries, Keys, and Values.

(b) Multi-Head-Attention Architecture, use Multi Queries (Q), Keys (K), and Values (V) to create the appropriate masks.

Figure 2.33: Attention Architecture

where γ is a learnable scalar parameter that initialized as 0 [220]. The sequence of outputs $\mathbf{y}_i$ has the same dimension as the input and $\mathcal{H} = \gamma Id$. To summarize, from now on, let us define the Self-Attention according to the following expression:

$$\mathbf{y}_i = \text{Attention}(\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_H, \mathbf{W}_V, \mathbf{x}_i, \mathbf{x}_j) \tag{2.41}$$

2.4.4 Multi-Head Attention

Multi-Head Self-Attention (MHSA), proposed initially by Vaswani *et al.* [219], was introduced for language models but found relevant utility in different application cases, such in seismology [46]. MHSA outperforms Self-Attention because rather than performing a single attention operation as in Equation 2.41, it extracts many queries, keys, and values from the same input.

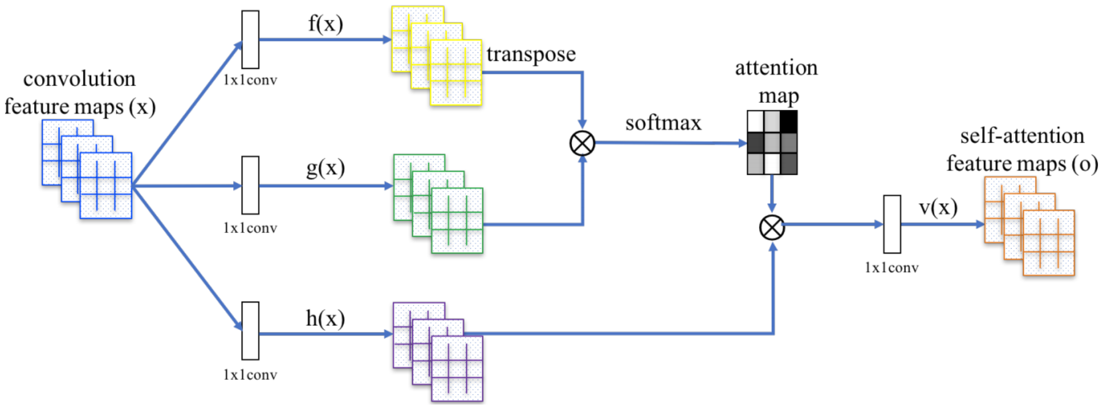


Figure 2.34: Self-Attention Module for designed for SAGAN in [220]

In doing so, MHSA introduces an intermediate linear embedding defined as:

$$\begin{aligned}\mathcal{H}_Q &: \mathbb{Q} \mapsto \mathbf{W}_Q^H \mathbb{Q} = \mathbb{H}_Q \\ \mathcal{H}_K &: \mathbb{K} \mapsto \mathbf{W}_K^H \mathbb{K} = \mathbb{H}_K \\ \mathcal{H}_V &: \mathbb{V} \mapsto \mathbf{W}_V^H \mathbb{V} = \mathbb{H}_V\end{aligned} \tag{2.42}$$

with $\mathbf{W}_Q^H \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_K^H \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_V^H \in \mathbb{R}^{d_{model} \times d_v}$. Usually $d_v = d_k = d_{model}/h$. MHSA forces the model to focus on different representation subspaces of the data at different positions [219]. In doing so, the intermediate embeddings in Equation 2.42 is passed to multiple parallel Attention layers (expressed by Equation 2.41) according to the expression:

$$\mathbf{head}_j = \text{Attention}(\mathbf{W}_Q^H, \mathbf{W}_K^H, \mathbf{W}_V^H, \mathbf{H}_i) \tag{2.43}$$

The Multi-Head attention adopts h parallel heads, as expressed in Equation 2.43, and the overall output can be schematically expressed as:

$$\mathbf{MultiHead}(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \mathbf{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h) W^O \tag{2.44}$$

Multi-head attention is the crucial mechanism for transformer-based architectures, presented in the following subsections and adopted in this thesis.

2.4.5 Transformer Architecture

The Transformer architecture is more complex, using Multi-Head Attention layers. In addition to that, this technique encompasses the Point-wise Feed-Forward Network (FFN), the positional encoding, embedding, encoder, and decoder modules (See the detail of the architecture in Figure 2.35). Transformers perform better for the following reasons :

- Multi-Head Attention is more complexes

In Transformer architecture, attention heads are concatenated and passed through FFN, which consists of two linear layers with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.45}$$

FFN is position-wise because using a feed-forward network modifies the representation at every position in the sequence.

The Positional Embedding essentially consists of sine and cosine functions. Calling pos the time step position and i the total dimension, the positional encoding reads:

$$\begin{aligned}\text{PosEnc}(pos, 2i) = &\quad \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ \text{PosEnc}(pos, 2i+1) = &\quad \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)\end{aligned} \tag{2.46}$$

The encoding wavelength follows a geometric progression from 2π to $1000 \cdot 2\pi$. Because the transformer includes MHSA, which analyses different portions of the sentences simultaneously, it becomes essential to identify which part of the sentence the attention is coming from so that a part analyzed should not overlap anotdher part. Also, that may allow the model to extrapolate the longer sequences for which the model has not been trained before during the

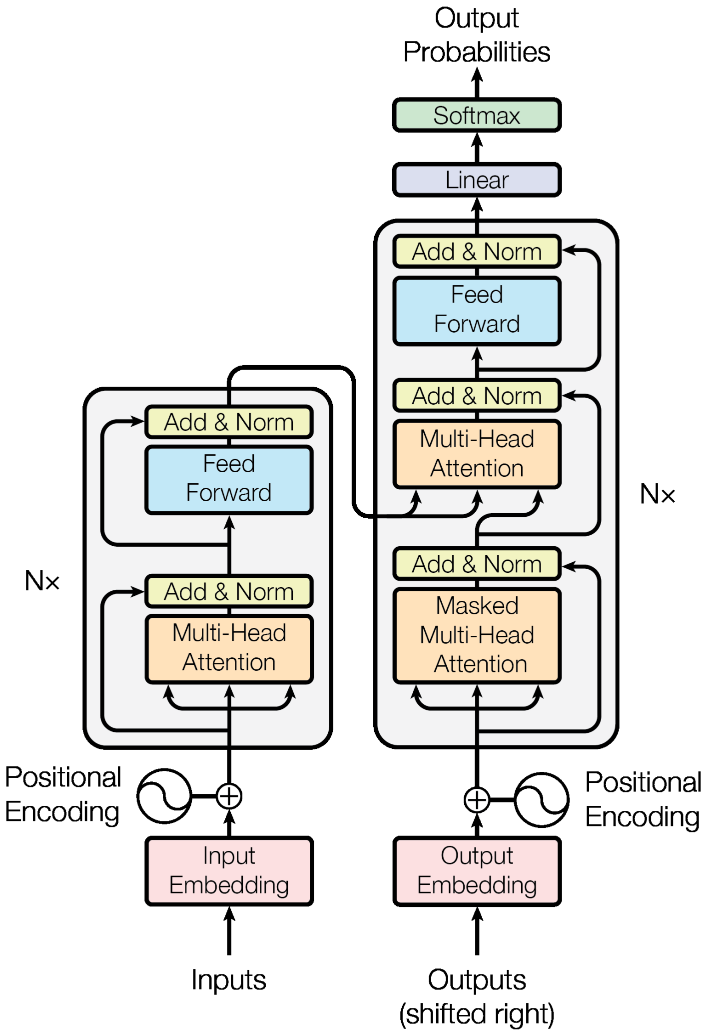


Figure 2.35: Transformer Architectures, courtesy of Vaswani *et al.*, [219]

training. Ideally, this concept can be adapted for earthquake signals, too. Another component often used with positional encoding is positional embedding. This latter injects context into the analysis of information sequences: each subsequence receives additional embedding based on its positional sequence. The type of embedding commonly used is called "vector embedding", which treats each subsequence as a vector of dimension d_{model}. According to Vaswani *et al.*, [219], positional embedding and positional encoding can be used in the same model without performance degradation. A transformer, more complex than an MHSA, preserves the causality of the input sequence thanks to position encoded [121]. This is even more effective for Conformers, as described in Section 2.4.6.

2.4.6 Conformer Architecture

The Conformer is a neural network developed in the context of audio speech recognition and separation that combines the advantages of transformers, § 2.4.5, such as the Attention Mechanism, § 2.4.4 above mentioned, with classical convolution layers [119]. This architecture represents an enhancement over previous models and has been proposed as a replacement for the Transformer by Gulati et al. [121]. According to the same authors, Conformer architecture is particularly effective in improving reconstruction and compared to Transformers:

- Transformers are less efficient at capturing minute and detailed features in a localized context.

- Transformers cannot capture and exploit local information and local connectivity.

- Conformers outperforms Transformers for audio tasks, Gulati *et al.*, 2020[121].

- Conformers combine the effectiveness of Transform to global view and the power of Convolutional neural network to local connectivity

A Conformer consists of four modules that are stacked upon each other. These modules include:

(i) First Feed-Forward Module, **FFN**: Unlike the previous definition in Section §2.4.5 above, the FFN module is a residual block comprising layer normalization, linear layers, Swish activation, and Dropout. Pre-norm residual units, at the opposite of post-norm, apply layer normalization immediately before the sublayer, [221]. For more details, refer to Figure 2.36c.

(ii) Multi-Head Self-Attention Modules, **MHSA**: This architecture, presented in subsection 2.4.4 (see more details in Figure 2.33b), has been introduced in Transformers-XL, Dai *et al.*, 2019 [222] and, as explained, is designed to capture long-range dependencies and ensures that the network remains robust to variations of the length of input values [121].

(iii) Convolution Modules, Conv: Unlike traditional convolutions used in image processing, Conformer employs a different convolution type. The structure respects the sandwich structure, inspired by Macaron-Net, [223]; it stacks LayerNorm, PointWise convolution, GLU activation, and Depth-wise Convolution and Swish activation functions. The LayerNorm is a type of normalization to the output of CNN or a Linear layer. The normalization has proceeded along the length of the signal. The point-wise convolution has a 1x1 kernel size, and then when the kernel spans, the signal length of each point is convoluted. It is used to view every signal time step of a signal.
The GLU or gated linear unit, developed in the framework of language modeling [224], is an activation function for an input $\mathbf{x}$, split into two halves part $\mathbf{a}$, $\mathbf{b}$ depending on the dimension required we observe:

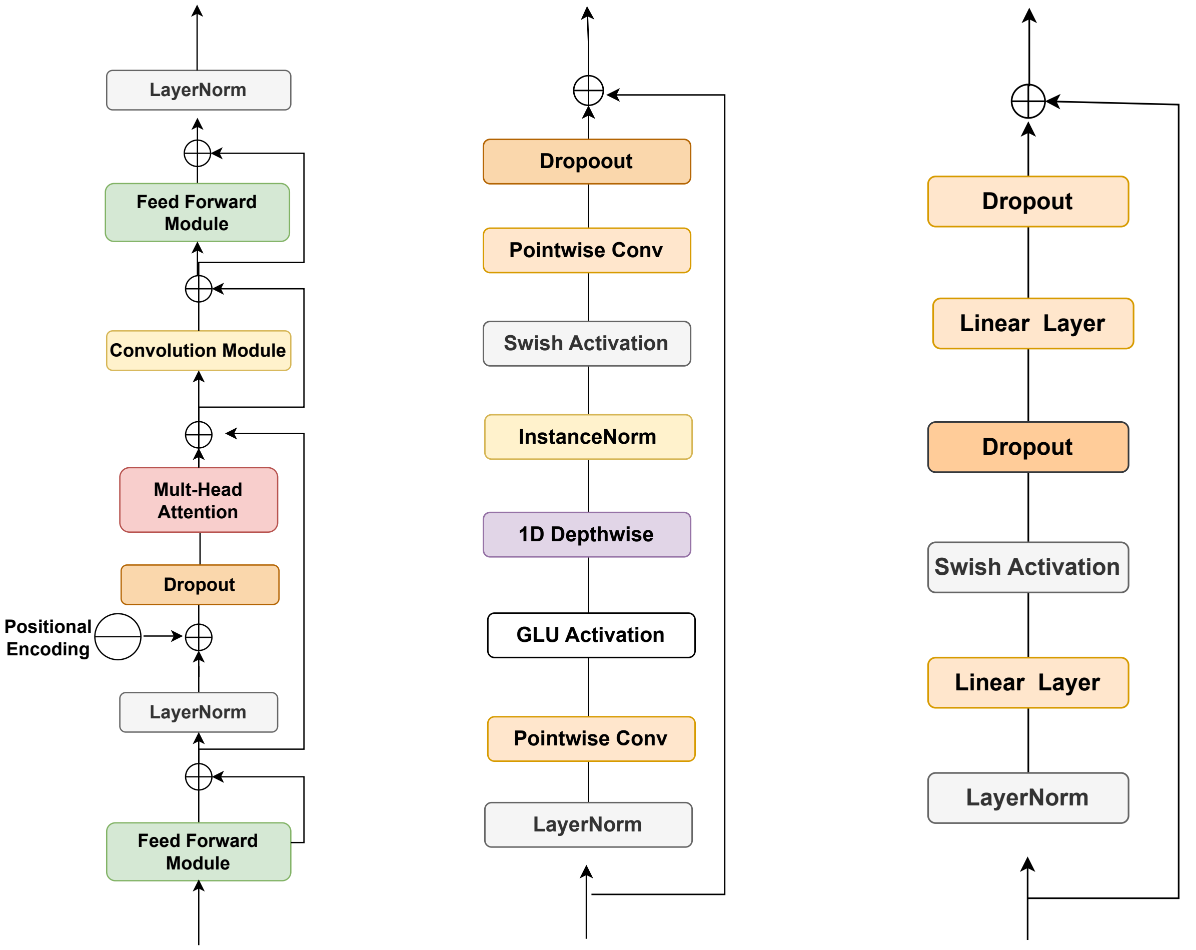
$$\mathrm{GLU}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \odot \sigma(\mathbf{b}) \tag{2.47}$$

where $\mathbf{x}$ and $\mathbf{y}$ are two inputs, $\odot$ an element-wise product between matrices.
The Swish activation function is also an activation function that is adopted because it is cheap to compute. According to Gulati *et al.*, 2020 [121], Swish activations lead to faster convergence in the Conformer models. The Swish activation reads:

$$\mathrm{Swish}(\mathbf{x}) = \mathbf{x} \odot \sigma(\mathbf{x}) \tag{2.48}$$

the symbol $\sigma(\cdot)$ is a sigmoid functions.

(iv) Second Feed-Forward Module: this module follows a similar structure to the first feed-forward module, providing additional processing and feature extraction. See more details about this architecture in Figure 2.36(c)



(a) Conformer Architecture (b) Convolutional Modules (c) Feed forward architecture

Figure 2.36: Conformer Architecture and different submodules adopted. Source Gulati *et al.*, 2020 [121]

A Conformer can be summarized according to the following sequence of operations:

$$\begin{aligned} \tilde{x}_i &= x_i + \tfrac{1}{2}\mathrm{FFN}(x_i) \\ x'_i &= \tilde{x}_i + \mathrm{MHSA}(\tilde{x}_i) \\ x''_i &= x'_i + \mathrm{Conv}(x'_i) \\ y_i &= \mathrm{LayerNorm}(x''_i + \tfrac{1}{2}\mathrm{FFN}(x''_i)) \end{aligned} \tag{2.49}$$

By combining these four modules, the Conformer architecture enables effective audio super-resolution and can adapt to different acceleration scenarios. However, it is crucial to consider the increased computational demands associated with this approach.

2.5 Distance minimization

In adversarial learning, the need to render reconstructed data from an embedding space or an auto-regressive encoder has prompted a focused exploration of “distance” minimization techniques between real and generated time histories. Nevertheless, how do we define a statistical “distance” between natural and synthetic AI-generated samples? This investigation within the domain of time series can be bifurcated into distinct categories. One aims to achieve optimal reconstruction in the time domain, while the other targets reconstruction within the frequency domain. Distance minimization is a crucial aspect of the machine learning tasks we have discussed, and we have explored various types of distance metrics applicable to 1D data. Defining the best model for a specific dataset involves defining the Empirical Risk Minimization (ERM). The ERM is an average of the loss values on the model. Perlaza *et al.*, 2022 [225], have investigated the ERM. From model set $\mathcal{M}$, a pattern set $\mathcal{X}$ and a label set $\mathcal{Y}$ We could extract a triplet $(\theta^*, X, Y) \in \mathcal{M} \times \mathcal{X} \times \mathcal{Y}$, which could define the optimal in relationship by a function f :

$$Y = f(\theta^*, X, Y)$$

The pattern is usually audio, images, time series, and videos. A label is often a number or set of numbers. The allusion here is made for classification, but as we will see in the line below, this could be referred to as an optimization problem.
The θ^* represent the optimal models. In we deal with estimated values. The goal is to minimize the distance between the optimal and approximate solutions. $(\hat{\theta}, u) \in \mathcal{M} \times \mathcal{X}$ such a way that $f(\hat{\theta}, u)$ is close to $f(\hat{\theta}, u)$. Therefore the empirical risk induced by a model θ with respect to a data set:

$$\mathbf{z} = ((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$$

is determine by the functions $\mathbf{L_z} : \mathcal{M} \to [0, +\infty)$, which satisfies :

$$\mathbf{L_z}(\theta)) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\theta, x_i), y_i) \tag{2.50}$$

Formula in which the risk function is difined by ℓ, where $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, +)$, as is it, the risk induced by the model θ is $\ell(f(\theta, x), y)$ with respect to labelled pattern $(x, y) \in \mathcal{X} \times \mathcal{Y}$, for the optimal model θ^* ℓ will be zero.
This, given $\mathbf{z}$, the θ will be considered better than a model $\mu \in \mathcal{M}$ if this inégality is satisfied : $\mathbf{L_z} < \mathbf{L_z}$. The ERM problem, therefore, is an optimization problem with the following formulation:

$$\min_{\mathbf{theta} \in \mathcal{M}} \mathbf{L_z}(\theta), \tag{2.51}$$

The solution si identified as :

$$\mathcal{T}(\mathbf{z}) \triangleq \arg \min_{\mathbf{theta} \in \mathcal{M}} \mathbf{L_z}(\theta) \tag{2.52}$$

The ground truth model $\theta^* \in \mathcal{T}(\mathbf{z})$ and $\mathbf{L_z}(\theta) = 0$. In practice, for time series applications, liner combinations of losses computed in time and frequency domains are often adopted, with different weights, for balanced training. On the one hand, combining a loss in the frequency domain with one in the time domain can benefit the reconstruction and generation of new time series without applying standard algorithms for images applied to spectrograms but using 1D neural architectures. However, experience shows that applying two loss functions that are too similar in the time domain, for example, may degrade the reconstruction or generation quality- this observation is true for the frequency domain, too. Furthermore, the effectiveness of these losses can be enhanced by incorporating feature matching and feature extractors into the training process. These principles, derived from practical experience, guide the thoughtful design and application of loss functions in machine learning tasks for 1D data. In the following, some common time- and frequency-based losses are described. Some of them will be adopted in this work. The following section will succinctly present and discuss these categories, shedding light on their objectives, assumptions, and ultimate conclusions in machine learning.

2.5.1 Temporal minimization distance

(i) **Euclidean distances**: Euclidean distances are defined by the norm distance between a designated input and its corresponding output generated by a neural network. The Euclidean distance is used as a loss. Commonly employed metrics include MAE (Mean Absolute Error) and MSE (Mean Square Error). If we define h_θ as a neural network, $\mathbf{x}_i$ the input sample and $\mathbf{y}_i$ the targeted sample, one can define the Equation to be satisfied by:

$$\mathcal{L}(h_\theta(\mathbf{x}), \mathbf{y}) = \frac{1}{N} \sum_{i=0}^{N} \|y_i - h_\theta(\mathbf{x}_i)\|^p \tag{2.53}$$

Formula in which the term $\mathbf{y}_i$ is the target signal. The term p represents the order of the ℓ-norm, ℓ^1 for MAE or ℓ^2 for MSE.

(ii) **Mean Squared Logarithmic Error**: When the time-series data have long-range data, the Mean Squared Logarithm Error (MSLE) could be used to address the reconstruction challenge. The MSLE computes the logarithm of the difference between the real and the predicted data according to the following expression:

$$\mathcal{L}_{\text{MSLE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N} \sum_{i=0}^{N} \left(\log(y_i + 1) - \log(\hat{y}_i + 1)\right)^2 \tag{2.54}$$

(iii) **Hyper-Spherical Loss** : The Hyper-Spherical Loss aims to minimize the angle between predicted and original time-series data, and it reads:

$$\mathcal{L}_{\text{HSL}} = \sum_{i=1}^{N} (1 - \cos(\theta_{\mathbf{y}_i, \hat{\mathbf{y}}_i}))^2 = \sum_{i=1}^{N} \left(1 - \frac{\|\mathbf{y}_i \cdot \hat{\mathbf{y}}_i\|}{\|\mathbf{y}_i\| \cdot \|\hat{\mathbf{y}}_i\|}\right)^2 \tag{2.55}$$

Hyper-Spherical Prototype loss (HSL) can enhance the stability of the minimization algorithm when combined with another temporal loss. Remarkably, the Hyper-Spherical Loss pairs effectively with the Focal Frequency loss, a topic we will delve into later in Section 2.5.2.

2.5.2 Time-series distance minimization in frequency domain

When dealing with time-series data, mere distance reduction may not suffice for optimal reconstruction. Authors like Jian *et al.*, 2021 and Abdlatif *et al.*, 2020 [226],[119] advocated that a better reconstruction quality is obtained by not only minimizing the euclidean distance in the time domain, but also in the frequency domain, considering both phase and amplitude information. This multifaceted evaluation provides a more comprehensive understanding of the effectiveness of the reconstruction process in the context of time-series analysis. The following presents some widely adopted loss functions to minimize the dissimilarity between two time-histories in the frequency domain.

(i) **Focal Frequency Loss**: Focal frequency drop (FFD), first designed to enhance the quality of pictures, has demonstrated its capability to perform good signal reconstruction. This cost function is computed in the Fourier domain, as explained by Jiang and his collaborators [226]. The main idea behind the loss is to weight the L^1-norm of the deviation between two signals, evaluated in the Fourier domain. In this sense, error reconstruction for frequency that is easier to match will have a lower weight. The generic FFL reads:

$$\mathcal{L}_{\text{FFL}} = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} w(u,v) \left| F_r(u,v) - F_f(u,v) \right|^2 \tag{2.56}$$

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{2.57}$$

In this formula,

- $f(x,y)$ is the pixel located at (x,y) coordinates of the image, of dimension $M \times N$.
- u and v represent the spatial coordinate of a frequency in the spectrum frequency spectrum.
- $w(u,v)$ represents the weight matrix, where $w(u,v) = |F_r(u,v) - F_f(u,v)|^\alpha$. The weight matrix undergoes normalization to ensure its values fall within the range$[0,1]$. In this normalization, a value of 1 corresponds to the frequency with the highest current and most lost frequency, emphasizing its significance, while the weights for easier frequencies are reduced.
- α is the tunable scaling factor to provide more or less flexibility to the training process ($\alpha = 1$ in our experiments).
- $F(u,v)$ is the complexe frequency value.
- $F_r(u,v)$ is the spatial frequency value of the real sample with the spectrum coordinate (u,v) with the expression $F_r(u,v) = a_r + b_r\,\mathrm{i}$ and the spatial frequency of the fake sample is designed as, $F_f(u,v) = a_f + b_f\,\mathrm{i}$.
- a and b are, respectively, the real and the imaginary parts.

The equivalent of FFL for a 1D signal would be is :

$$\text{FFL} = \frac{1}{N} \sum_{u=0}^{N} w(u) \left| F_r(u) - F_f(u) \right| \tag{2.58}$$

expression in which :

- t is the instant in the time signal.
- u is the value from the spectrum
- $x(t)$ is the value of the signal at time t.
- $F(u) = \sum_{i=1}^{N-1} x(t) e^{i2\pi\left(\frac{u \cdot t}{N}\right)}$
- $w(u) = |F_r(u) - F_f(u)|^\alpha$
- $F_r(u) = a_r + b_r\mathrm{i}$
- $F_f(u) = a_f + b_f\mathrm{i}$

Implementing the FFL for time series was used in our network architecture to improve signal reconstruction quality. See the illustration of the FFL loss in Figure 2.37.

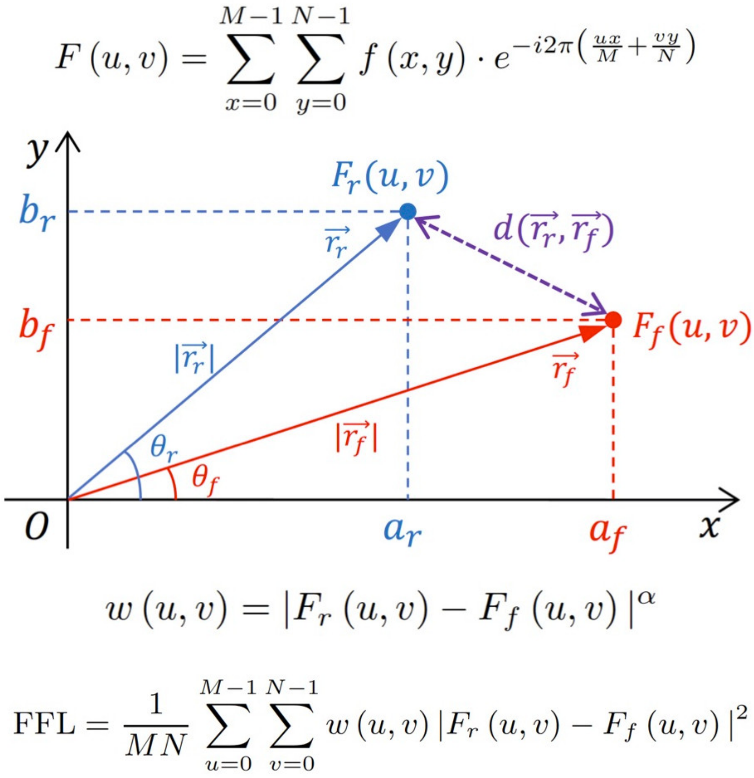


Figure 2.37: Focal frequency Loss from the Complexe domain. Source: [226].

(ii) **MelSpectrogram**: MelSpectrogram is a distance function based on the signals' spectrograms, which "*is defined as the magnitude component of the short-time Fourier transform (STFT) of an input waveform.* The MelSpectrogram uses the Mel Scale factor, a perceptual scale of pitches judged by human listeners to be equal in distance from one another. The Mel Scale adopts as a reference point a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listener's threshold. Above about 500 Hz, increasingly large intervals are judged by listeners to produce equal pitch increments. In particular, this scale value is :

$$M = 2595\log_{10}\left(1 + \frac{f}{700}\right) \tag{2.59}$$

A loss function for the MelSpectrogram can be defined. The MelSpectrogram projects information from time-series data into a latent space, enabling the manipulation of information present in both the time and frequency domains. The process involves several steps. First, the time series is pre-processed, i.e., divided into short overlapping frames. In the subsequent step, each frame is multiplied by a window to reduce spectral leakage, known as windowing. Following this, a Fast Fourier Transform (FFT) is applied to each windowed frame, providing its representation in the Fourier space. In the penultimate step, the logarithm of the filter bank energies is taken to compress the dynamic range. Finally, a cosine filter is applied to decorrelate the filter bank further and extract salient features, as shown in Equation 2.60:

$$\mathcal{L}_{\mathcal{S}}(\mathbf{y},\hat{\mathbf{y}}) = \sum_{s\in 2^6,\ldots 2^{11}} \sum_t \|\mathcal{S}_t^s(y)) - \mathcal{S}_t^s(\hat{\mathbf{y}}))\|_1 + \alpha_s \sum_t \|\log\mathcal{S}_t^s(y)) - \log\mathcal{S}_t^s(\hat{\mathbf{y}}))\|_2 \tag{2.60}$$

Expression in which the term $\mathcal{S}_t^s(\cdot)$ represents the Mel-Spectrogram, while t-th fram of 64-bin s represents the length of the windows. We have $\alpha_s = \sqrt{s/2}$.

The loss $\mathcal{L}_{\mathcal{S}}(\mathbf{y}, \hat{\mathbf{y}})$ in Equation 2.60 has been widely adopted to capture long-range dependencies and achieve the high-fidelity reconstruction of data (see Zeghidour *et al.*, 2021 [227] for SoundSteam [227]).

(iii) **Short-Time Fourier Transform** Short-Time Fourier Transform (STFT) is largely adopted to train advanced neural architecture for speech recognition, *i.e.*, to predict raw, continuous speech waveform samples because loss functions can be reformulated through this STFT operator. The proposed loss is computed by considering both amplitude and phase information, thereby incorporating essential elements for accurately enhancing the model's ability to capture and reproduce the intricacies of speech waveforms. For instance, a variant of Equation 2.60 has been proposed Kreuk *et al.*, 2023 [228], for training the model MusicGen:

$$\mathcal{L}_{\mathcal{S}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{|\alpha| \cdot |s|} \sum_{\alpha_i \in \alpha} \sum_{i \in e} \left\| \mathcal{S}_i(\boldsymbol{y}) - \mathcal{S}_i(\hat{\boldsymbol{y}}) \right\|_1 + \alpha_i \left\| \mathcal{S}_i(\boldsymbol{y}) - \mathcal{S}_i(\hat{\boldsymbol{y}}) \right\|_2 \tag{2.61}$$

In this formula, $\mathcal{S}_i$ is a 64-bin mel-spectrogram using a normalized STFT with window size of 2^i and hop length of $2^i/4$, $e = 5, \ldots, 11$ is the set of scales. The term α represents the scalar coefficients balancing the L1 and L2 terms; the α_i equals 1. If we design a STFT complex spectral sequence $\mathbf{Y} = \left[\mathbf{Y}_1^\top, \ldots, \mathbf{Y}_T^\top\right] \top$ is represented by using a matrix $\mathbf{W}$ as follow:

$$\mathbf{Y} = \mathbf{W}\mathbf{y} \tag{2.62}$$

Where t represents the frame index, $\mathbf{W}$ represents a matrix which performs STFT operation. the term n represents the bin at frame t. According to Shinji Takaki et *al.*, 2018 [229], the formulation for amplitude :

$$\begin{aligned} Y_{t,n} &= \boldsymbol{W}_{t,n}\boldsymbol{y}, \\ A_{t,n} &= |Y_{t,n}| \\ &= \left(\boldsymbol{y}^\top \boldsymbol{W}_{t,n}^H \boldsymbol{W}_{t,n}\boldsymbol{y}\right)^{\frac{1}{2}} \end{aligned} \tag{2.63}$$

In addition to the formulation for the amplitude, an equivalence could be found for the phase; the corresponding expression is the following equation :

$$\begin{aligned} \exp\left(i\theta_{t,n}\right) &= \exp\left(i\angle Y_{t,n}\right) \\ &= \frac{Y_{t,n}}{A_{t,n}} \\ &= \frac{\boldsymbol{W}_{t,n}\boldsymbol{y}}{\left(\boldsymbol{y}^\top \boldsymbol{W}_{t,n}^H \boldsymbol{W}_{t,n}\boldsymbol{y}\right)^{\frac{1}{2}}} \end{aligned} \tag{2.64}$$

Also, the equation for the amplitude loss evaluates the difference between the square values of the amplitude values provided by the STFT and the predicted output values:

$$\begin{aligned} E_{t,n}^{(amp)} &= \frac{1}{2}\left(\hat{A}_{t,n} - A_{t,n}\right)^2 \\ &= \frac{1}{2}\left(\hat{A}_{t,n} - \left(\boldsymbol{y}^\top \boldsymbol{W}_{t,n}^H \boldsymbol{W}_{t,n}\boldsymbol{y}\right)^{\frac{1}{2}}\right)^2 \end{aligned} \tag{2.65}$$

The phase spectrum is a periodic variable with a period of 2π. To account for this periodic property, a loss function for the phase spectrum at frequency bin n and frame t is defined as follows:

$$\begin{aligned} E^{(ph)}_{t,n} &= \frac{1}{2}\left|1 - \exp\left(i(\hat{\theta}_{t,n} - \theta_{t,n})\right)\right|^2 \\ &= 1 - \frac{1}{2}\left(\frac{\hat{Y}_{t,n}}{\hat{A}_{t,n}}\frac{\left(\boldsymbol{y}^\top \boldsymbol{W}^H_{t,n}\boldsymbol{W}_{t,n}\boldsymbol{y}\right)^{\frac{1}{2}}}{\boldsymbol{W}_{t,n}\boldsymbol{y}} + \frac{\overline{\hat{Y}}_{t,n}}{\hat{A}_{t,n}}\frac{\left(\boldsymbol{y}^\top \boldsymbol{W}^H_{t,n}\boldsymbol{W}_{t,n}\boldsymbol{y}\right)^{\frac{1}{2}}}{\boldsymbol{W}_{t,n}\boldsymbol{y}}\right) \end{aligned} \tag{2.66}$$

And the complete loss to optimize is :

$$E^{(sp)} = \sum_{t,n}\left(E^{(amp)}_{t,n} + \alpha_{t,n}E^{(ph)}_{t,n}\right) \tag{2.67}$$

(iv) **Wavelet Transform**: Employing wavelet transform to enhance the quality of generation or reconstruction is not novel. Various studies have demonstrated that this decomposition method can effectively tackle the challenge of high-frequency generation, as indicated by Lukas Prantl *et al.*, 2022. Numerous variants of this loss exist, ranging from distance matching to adversarial computation. Despite the diversity in implementation, the fundamental idea remains consistent in decomposing information at different levels to extract relevant features.

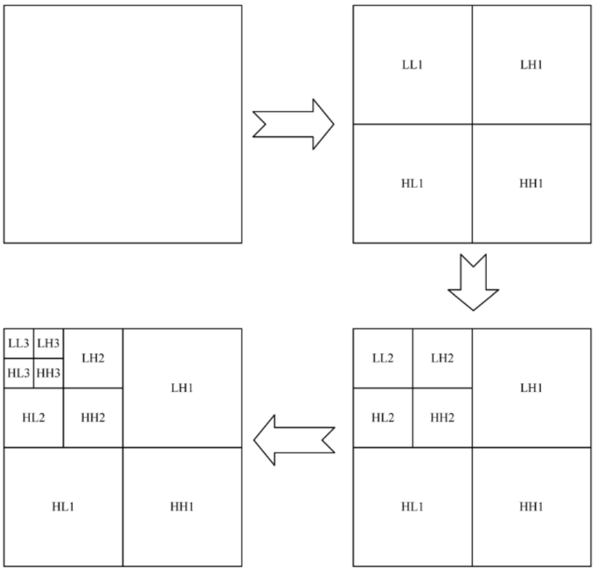


Figure 2.38: The process of wavelet decomposition of a data, according to Qiuyu Zhu *et al.*, 2021 [insert citation]

According to the Qiuyu Zhu *et al.*, 2021 [230], in the initial stage of wavelet transformation, information is partitioned into LL1, LH1, HL1, and HH1. The low frequency, denoted as LL1, can be perceived as the envelope for time series or corresponds to the contour and shape of images. LH1, HL1, and HH1 represent high-frequency components. Subsequently, at the second level, the transformation is applied to the low-frequency part, LL1, resulting in LL2, LH2, HL2, and HH2, view Figure 2.38. This process reveals a pyramid-like structure in the wavelet representation. The LL component of the last level concentrates the primary energy in the audio signal. Researchers have noted that the wavelet loss function emphasizes discerning differences between frequencies. After performing a multi-level wavelet transform, various frequency characteristics at different spatial positions can be obtained. To produce precise data, it is essential to constrain both high and low-frequency components. However, restricting only the high-frequency components may result in insufficient details, leading to noisy output data. To address this, researchers propose constraining the modulus of high-frequency information. Thus, the equations that need to be satisfied are:

$$\begin{aligned}
\mathcal{L}_1 &= \lambda_{11}\sum_{i=1}^{n}(m-m^*) + \lambda_{12}\sum_{i=1}^{n}|h_1-h_1^*| \\
\mathcal{L}_2 &= \sum_{i=1}^{n}|h_2-h_2^*| \\
\mathcal{L}_3 &= \sum_{i=1}^{n}|h_3-h_3^*| \\
\mathcal{L}_4 &= \sum_{i=1}^{n}(l_2-l_2^*)
\end{aligned} \tag{2.68}$$

In those formulations, the terms are :

- l_1 and h_i refer to the low-frequency component (LL) and the high-frequency components (LH, HL, HH) after the i-th wavelet-transform.
- m represents the element-by-element modulus of the high-frequency components.
- n is the number of samples
- the λ_j are the trade-off parameters.

Then the total loss to be satisfied is:

$$\mathcal{L}_{\text{wav}} = \sum_{j=1}^{4}\lambda_j\mathcal{L}_j \tag{2.69}$$

2.5.3 Feature Matching

Feature matching involves extracting features from input data using a discriminator neural network. This approach is commonly employed in generative models, where the generation quality is improved through discriminator training (see Section 2.3.3). Various features are extracted at different levels of the discriminator's architecture for both the samples in the data set and the generated samples to align their distributions closely. The distances between these features are minimized using an ℓ^1 loss. Intuitively, this process can be interpreted as a learned similarity metric, where the discriminator learns a feature space that discriminates between "fake" and "real" data. It is essential to notice that there is no use of any loss in the raw audio space, distinguishing it from other conditional GANs where L1 loss is applied to match conditionally generated images with their corresponding ground truths, enforcing global coherence (Kundan Kumar *et al.*, 2019[231]). An example of feature extraction of time series, for instance, is provided by the following expression:

$$\mathcal{L}_{\text{FM}}(G,D) = \mathbb{E}_{\mathbf{y},\mathbf{s}\sim p_{\text{data}}}\left[\sum_{i=1}^{T}\frac{1}{N_i}\left\|D^{(i)}(\mathbf{y}) - D^{(i)}(G(\mathbf{s}))\right\|_1\right] \tag{2.70}$$

In this formula, the term $D^(i)$ represents the output of the i-th layer of the discriminator.

Name	$D_f(P \Vert Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x$	$u\log u$	$1+\log\frac{p(x)}{q(x)}$
Reverse KL	$\int q(x)\log\frac{q(x)}{p(x)}\mathrm{d}x$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson $\mathcal{X}^2$	$\int\frac{(q(x)-p(x))^2}{p(x)}\mathrm{d}x$	$(u-1)^2$	$2\left(\frac{p(x)}{q(x)}-1\right)$
Squared Hellinger	$\int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\mathrm{d}x$	$(\sqrt{u}-1)^2$	$\left(\sqrt{\frac{p(x)}{q(x)}}-1\right)\sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2}\int p(x)(\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\mathrm{d}x)$	$-(u+1)\log\frac{1+u}{2}+u\log u$	$\log\frac{2p(x)}{p(x)+q(x)}$
GAN	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\mathrm{d}x-\log(4)$	$u\log u - (u+1)\log(u+1)$	$\log\frac{p(x)}{p(x)+q(x)}$

Table 2.6: This is the list of f-divergence $D_f(P \Vert Q)$ used in adversarial learning [232]

2.5.4 Feature Extractors

Feature extractors are pre-trained network architectures specifically designed to extract relevant features. They are not limited to generative tasks and can also be applied to auto-regressive models. The weights of feature extractor models, often frozen at training time, are typically derived from a network trained for classification tasks. In such cases, having a benchmark data set is essential. In seismology, overcoming this challenge is addressed by utilizing phase picking. The phase-picking process helps detect the general patterns of a seismogram, including factors such as consistent arrival times of signals and phase attenuation. Notably, EQTransformers by Mousavi have been found to perform effectively in this context, as detailed in Section 2.1.3.

2.5.5 f-divergence distances

f-divergence is the class of statistical divergence and a generalization of the well-known *Kullback-Leibler divergence* [232, 164]. Given two distribution P and Q that posses, respectively, an absolutely continuous density function p and q with respect to a base Lebesgue measures dx defined on the domain $\mathcal{X}$, their f-divergence reads:

$$D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \tag{2.71}$$

where the generator function $f : \mathbb{R}_+ \to \mathbb{R}$ is a convex, lower-semi-continuous function stratifying $f(1) = 0$. Different choices of f are possible, as shown in Table **??**.

The variational estimations of those functions are given by :

$$D_f(P\|Q) = \sup_{T\in\mathcal{T}} \left(\mathbb{E}_{x\sim P}[T(x)] - \mathbb{E}_{x\sim Q}[f^*(T(x))]\right) \tag{2.72}$$

formula in which $\mathcal{T}$ is an arbitrary class of $T : \mathcal{X} \to \mathbb{R}$. The function T is defined in such a way that :

$$T^*(x) = f'\left(\frac{p(x)}{q(x)}\right) \tag{2.73}$$

f' denotes the first-order derivative of f. This condition can serve as a guiding principle for the choice of f. To estimate a generative model Q given a true distribution, P, we could

Name	Output activation g_f	dom_{f^*}	Conjugate $f^*(t)$	$f'(1)$
Kullback-Leibler (KL)	v	$\mathbb{R}$	$\exp(t-1)$	1
Reverse KL	$-\exp(-v)$	$\mathbb{R}_-$	$-1-\log(-t)$	-1
Pearson $\mathcal{X}^2$	v	$\mathbb{R}$	$\frac{1}{4}t^2+t$	0
Squared Hellinger	$1-\exp(-v)$	$t<1$	$\frac{t}{1-t}$	0
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$t<\log(2)$	$-\log(2-\exp(t))$	0
GAN	$-\log(1+\exp(-v))$	$\mathbb{R}_-$	$-\log(1-\exp(t))$	$-\log(2)$

Table 2.7: This table summarizes the recommended final layer activation functions and critical variational function levels defined by $f'(1)$. The value $f'(1)$ can be interpreted as a classification threshold applied to $T(x)$ to distinguish between true and generated samples [232].

parameterize Q through a set of weights θ, i.e., through Q_θ. We use T as a variational function, parameterized by a vector ω and written, T_ω.

$$F(\theta,\omega) = \mathbb{E}_{x\sim P}[T_\omega(x)] + E_{x\sim Q_\theta}[-f^*(T_\omega(x))] \tag{2.74}$$

A connection could be made with f-divergences, according to Nowozin *et al.* [232], if the domain dom_f^* of the conjugate functions f^*x is respected. If we replace $T_\omega = g_f(V_\omega(x))$ and rewrite the saddle objective we obtain :

The general expression becomes:

$$F(\theta,\omega) = \mathbb{E}_{x\sim P}[g_f(V_\omega(x))] + E_{x\sim Q_\theta}[-f^*(g_f(V_\omega(x)))] \tag{2.75}$$

where $V_\omega : \mathcal{X} \to \mathbb{R}$ and $g_f : \mathbb{R} \to \mathrm{dom}_{f^*}$ is an output activation function specific to the f-divergence used.

As an example, in the case of f-GAN, we have:

$$F(\theta,\omega) = \mathbb{E}_{x\sim P}[\log(D_\omega(x))] + \mathbb{E}_{x\sim Q_\theta}[\log(1-D_\omega(x))] \tag{2.76}$$

2.6 Reconstruction metrics based on time-frequency misfit

The Kristeková Goodness-of-Fit Criteria

In signal processing, particularly in assessing signal reconstruction quality, the imperative arises to establish a methodology for analyzing disparities between signals, specifically in the context of three-component seismic signals. While a cursory visual examination may afford a preliminary assessment of reconstruction quality, the exigency for quantification mandates using a judicious mathematical framework to address the misfit dilemma systematically.

A straightforward approach involves a natural observation of the disparity between the signal $s(t)$ and the reference signal $s_r(t)$, expressed as $D(t) = s(t) - s_r(t)$. Regardless, while providing an estimate of signal dissimilarity, this expression is inherently insufficient in characterizing the nature or origin of the observed distinctions. Notably, this simplistic representation cannot discern frequency, amplitude, or phase variations across temporal and spectral domains. Recognizing the distinctiveness of observations in the time and frequency domains, exploring differences in the time-frequency domain has become customary. The most pertinent technique for such analysis is the Time-Frequency misfit criteria, which concurrently scrutinizes discrepancies at each (t, f) point.

Consequently, Kristeková et al. (2006) [233] refined the development of time-frequency envelope and phase misfit criteria, demonstrating their efficacy in quantifying and characterizing comprehensive differences between signals. Two main assumptions are behind the choice of this metric: one of the two signals can be viewed as some modification of the other sigma, and the most complete and informative characterization of a signal can be obtained by its time-frequency representation, rendered through continuous wavelet transform. In the formulation of a continuous wavelet transform of a signal $s(t)$ is defined by the following expression:

$$CWT_{(a,b)}\{s(t)\} = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} s(t)\phi^* \left(\frac{t-b}{a}\right) dt \tag{2.77}$$

with t being time, $a(f)$ a frequency-dependent scale parameter, proportional to the frequency f; b is the translation parameter, and $\phi : \mathbb{R} \to \mathbb{R}$ a Morlet wavelet expressed as follows:

$$\phi : t \mapsto \pi^{1/4} \exp(\mathrm{i}\omega_0 t) \exp(-t^2/2) \tag{2.78}$$

with $\omega_0 = 6$ being the proper choice for a wide variety of seismic signals [234]. Finally, the The form of the equation that is used is the

$$W(t,f) = CWT_{(f,t)}\{s(t)\} = \sqrt{\frac{2\pi|f|}{\omega_0}} \int_{-\infty}^{\infty} s(\tau)\phi^* \left(2\pi f \frac{\tau - t}{\omega_0}\right) \tag{2.79}$$

The $W^2(t,f)$ represents the energy distribution of the signal in the Time-Frequency plan. The envelope is noted : $A(t,f)$ and the phase is noticed $\phi(t,f)$, such a way that :

$$A(t,f) = |W(t,f)|\,, \phi(t,f) = \mathrm{Arg}[W(t,f)] \tag{2.80}$$

Consider a signal $s(t)$ and a reference signal $sr(t)$; the envelope of the reconstruction is given by

$$\Delta A(t,f) = A(t,f) - Ar(t,f) = |W(t,f)| - |W_r(t,f)| \tag{2.81}$$

The Equation 2.6 is the difference at each (t,f) point. Similarly, for the phase-in range $[-\pi,\pi]$:

$$\Delta\phi(t) = \mathrm{Arg}\left[\frac{W(t,f)}{W(t,f)}\right] \tag{2.82}$$

However, for time-local comparison between two signals, the difference between amplitudes is normalized by the amplitude of the reference signal. The phase values are instead normalized by ϕ.

$$\mathrm{TFEM}_{\mathrm{LOC}}(t,f) = \frac{\Delta A(t,f)}{A_r(t,f)} \tag{2.83}$$

$$\mathrm{TFPM}_{\mathrm{LOC}}(t,f) = \frac{\Delta\phi(t,f)}{\pi} \tag{2.84}$$

The $\mathrm{TFEM}_{\mathrm{LOC}}$ define the locally normalized TF envelope misfit criteria, and the $\mathrm{TFPM}_{\mathrm{LOC}}(t,f)$ define the locally normalized phase criteria. For the global misfit classification, Kristeková *et al.* proposed to use the following formulations :

$$\begin{aligned}
\mathrm{TFEM}_{\mathrm{GLOB}}(t,f) &= \frac{Ar(t,f)}{\max_{t,f}\{Ar(t,f)\}} \mathrm{TFEM}_{\mathrm{LOC}}(t,f) \\
&= \frac{\Delta A(t,f)}{\max_{t,f}\{Ar(t,f)\}}, \\
\mathrm{TFPM}_{\mathrm{GLOB}}(t,f) &= \frac{Ar(t,f)}{\max_{t,f}\{Ar(t,f)\}} \mathrm{TFPM}_{\mathrm{LOC}}(t,f) \\
&= \frac{Ar(t,f)}{\max_{t,f}\{Ar(t,f)\}} \frac{\Delta\phi(t,f)}{\pi}.
\end{aligned} \tag{2.85}$$

Coodness-of-Fit	
Verbal Value	**Numerical Value**
Excellent	8-10
Good	6-8
Fair	4-6
Poor	0-4

Table 2.8: Verbal representation of the discrete G.O.F score

It is often very useful to have a single-values envelope and phase misfits, $\mathrm{EM}_{\mathrm{GLOB}}$ and $\mathrm{PM}_{\mathrm{GLOB}}$.

$$\begin{aligned} \mathrm{EM}(t,f) &= \sqrt{\frac{\sum_f \sum_t |W_{ri}|^2 |TFEM_{\mathrm{LOC},i}(t,f)|}{\max_i \left(\sum_f \sum_t |W_{ri}|^2\right)}} \\ \mathrm{PM}(t,f) &= \sqrt{\frac{\sum_f \sum_t |W_{ri}|^2 |TFPM_{\mathrm{LOC},i}(t,f)|}{\max_i \left(\sum_f \sum_t |W_{ri}|^2\right)}} \end{aligned} \tag{2.86}$$

According to Kristekova, globally normalized misfit criteria are extremely useful in earthquake ground motion analysis and earthquake engineering, where there is a poor interest in relatively "small" magnitudes. This is why this metric has been chosen for this work. The TF envelope goodness-of-fit criteria can be introduced based on the TF envelope misfits :

$$\begin{aligned} \mathrm{TFEG}(t,f) &= A \exp\{-\left|TFEM(t,f)^k\right|\} \\ \mathrm{EG}(t,f) &= A \exp\{-\left|EM(f)^k\right|\} \\ A > 0,&\ k > 0 \end{aligned} \tag{2.87}$$

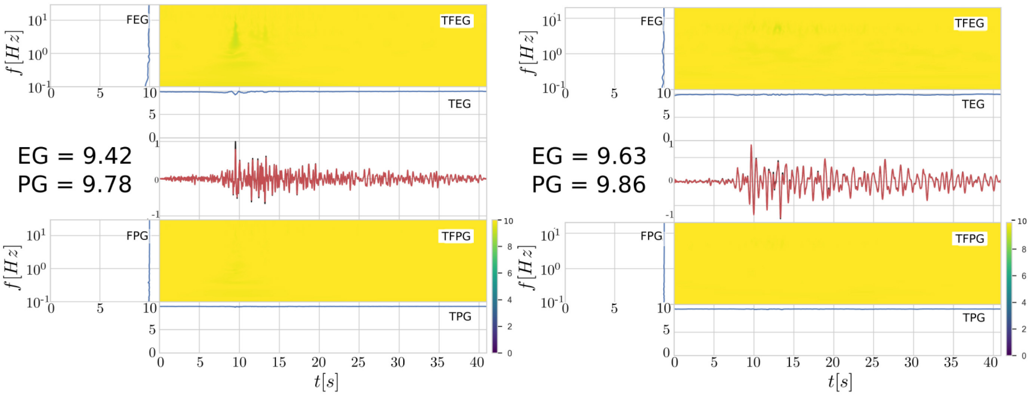
We could define the goodness-of-fit criteria as the goodness-of-fit equivalents to the TF phase misfit criteria :

$$\begin{aligned} \mathrm{TFPG}(t,f) &= A\left(1 - |\,\mathrm{TFPM}(t,f)|^k\right) \\ PG &= A\left(1 - |PM|^k\right). \end{aligned} \tag{2.88}$$

Generally, $A = 10$ and $k = 1$. From a seismological standpoint, if the score is between 9 and 10, the reconstruction quality is considered "excellent", whereas score values between 7 and 8 are considered a "good" fit. Other score values range from fair (4-6) to poor (<4), as listed in Table 2.8. See Figure 2.39. In summary, the misfit criteria utilizing wavelet transform enables the quantification of differences between distinct signals or three-component signals. Specifically tailored to time histories, this criterion is adopted to capture nuanced variations. Our analysis will consistently monitor the Goodness-of-Fit coefficient, EG, and PG components to assess the machine learning model's performance.

2.7 Database of Seismology

Recent developments in Machine Learning applied in seismology have shown the need for relevant and accurate data to process the task. It has been demonstrated that for a model or algorithm to be robust, the availability of a large amount of seismic data and computer resources must be promoted. This accessibility will enable the model to perform better when processing and analyzing seismic data. Even though labeled data exists in Seismology, the



(a)

(b)

Figure 2.39: Different values of GOF for the signals. Source: Gatti *et al.*, 2020 [21]

reliability of these labels is highly variable or somehow is not present in the database. A lack of high-quality labeled data sets could be a challenge to serve as ground, and the lack of standard benchmarks presents an obstacle to more rapid progress. Also, an enormous amount of seismic data is captured daily, and much of that ground motions is due to sources other than earthquakes and human activities, which are referred to as *non-earthquake* signals. In addition to the difficulty of getting access to seismic data, we should consider that a large and high-quality-label benchmark data set for seismic data does not exist. “In the absence of a standard benchmark, authors set their own criteria for evaluating performance. This inhibits progress because it makes it difficult to determine the relative performance, as well as the advantages and weaknesses, of each method.” [235]. Therefore, constructing a large-scale database with decent labels is challenging to address. The solutions start with The STanford EArthquake Data set (STEAD). This represents the foremost high-quality, large-scale global data set encompassing earthquake and non-earthquake signals recorded by seismic instruments. Two categories of data are present: the local earthquake waveforms (recorded at “local” distances within 350 km of earthquakes) and the seismic noise waveform that is free of the earthquakes. These

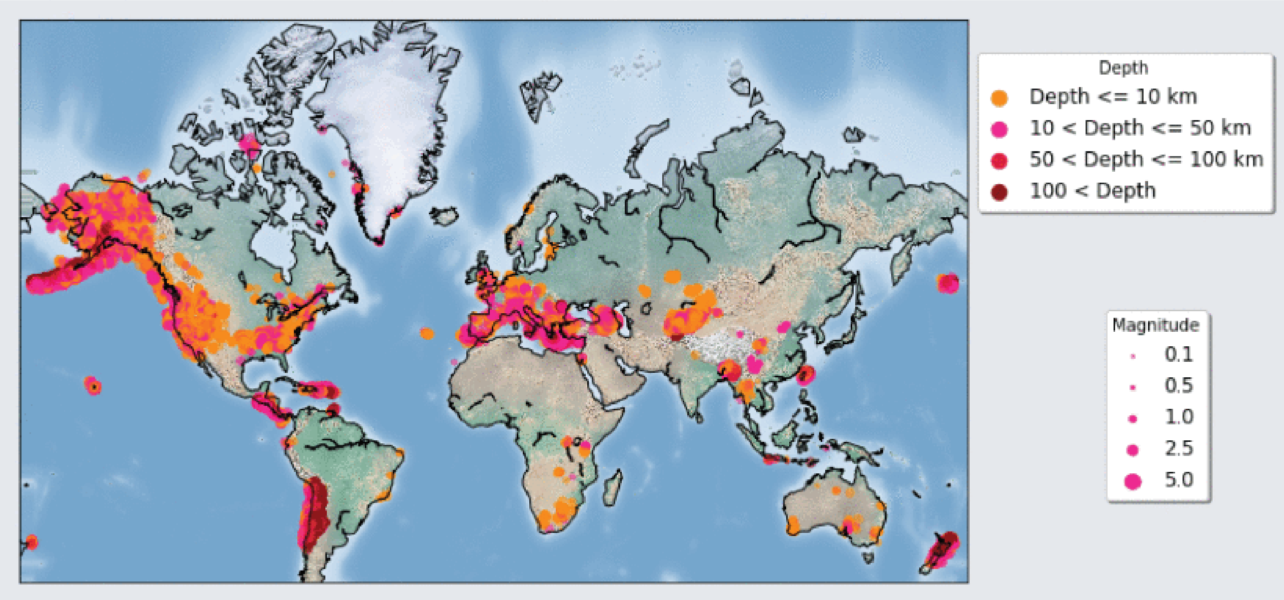


Figure 2.40: Localisation, size and depth distribution of recorded earthquakes. Source [235]

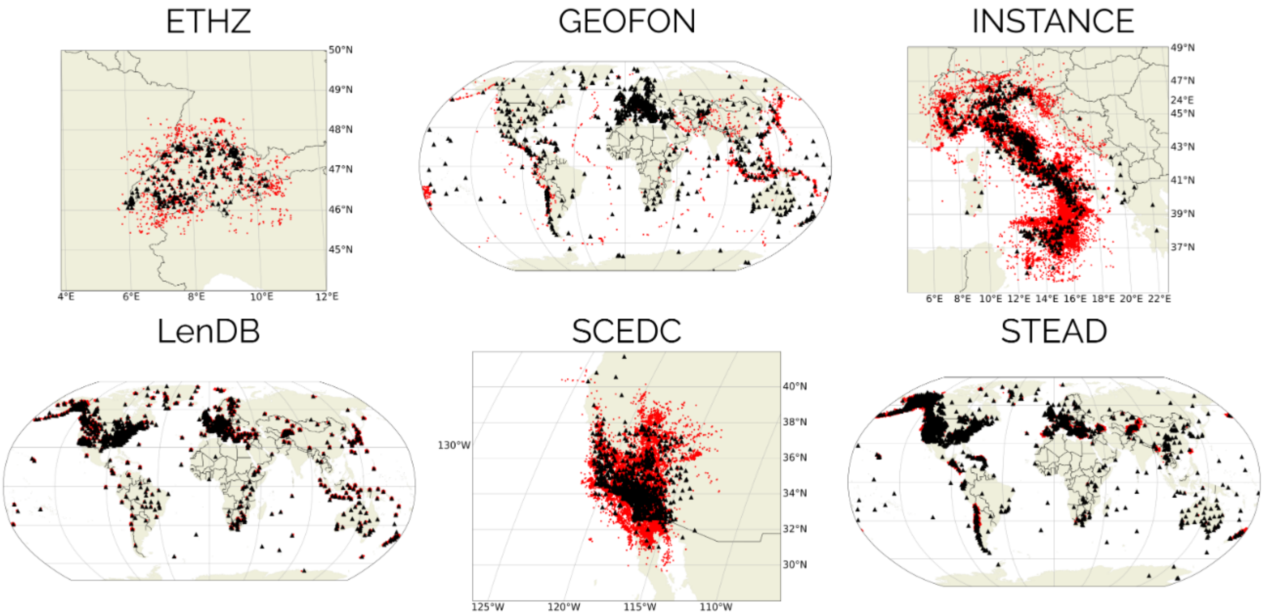


Figure 2.41: Benchmark data sets of seismic data. Illustration from [51]

data comprise $\sim$1.2 million time-series or more than 19,000 hours of seismic signal recordings between January 1984 and August 2018.

According to Mousavi *et al.*, 2020 [235], the structure of the STEAD can accelerate the progress in applying machine learning to problems in seismology. This should facilitate training, validation, and performance comparison. The database is available on `https://github.com/smousavi05/STEAD`. In the STEAD, earthquakes are typically captured using three-component instruments (seismographs) equipped with one vertical and two orthogonal horizontal sensors to characterize the vector components of ground motion. The sensors have accelerograms that detect horizontal motions in the direction North-East, East-West, and vertical motion Up-Down. “[. . .]The earthquake class contains only one category of local earthquakes with about 1050000 three-component time-histories (each 1 minute long) associated with $\sim$450000 earthquakes between January 1984 and August 2018. The earthquakes in the data set were recorded by 2613 receivers (seismometers) worldwide located at local distances (within 350 km of the earthquakes).”, see Figure 2.40.

Inclusive of waveforms, the data set encompasses essential metadata. This metadata comprises parameters such as magnitude, depth, recording stations, recorded earthquakes, and manually selected details such as the arrival times of P and S waves at each station. The data acquisition is sourced from entities such as the International Seismological Center and the National Earthquake Information Center, as reported by Mousavi et al. (2019) [235]. Ultimately, this compilation involves extracting and reorganizing metadata associated with local waveforms, amounting to 120 million data entries from these diverse resources. In conclusion, the fusion of Machine Learning and seismology necessitates reliable data, and STEAD stands out as the inaugural database to effectively address this challenge. Designed as a benchmark for machine learning tasks, STEAD provides a global, high-quality data set with over 1.2 million seismic recordings. Its impact is evident in inspiring the creation of other databases like ETHZ, GEOFON, INSTANCE, LenBD, and SCEDC [51]. View Figure 2.41. In our thesis, we leverage the comprehensive STEAD database due to its breadth, aligning perfectly with the machine learning tasks.

2.8 Partial conclusions and proposals of the thesis

In this chapter, we have reviewed the machine learning tasks in the realm of seismology before we tackle the problem of enhancing the result of physic-based numerical earthquake simulation in the pursuit of the original seminal work of Paolucci *et al.* [85] and Gatti and Clouteau, 2020 [21].

Machine learning in seismology was first introduced for earthquake discrimination. The accomplishment and progress have permitted other researchers to pay attention to artificial intelligence. Typical use cases go from simple classification tasks to Early Earthquake Warning (EEW). Subsequently, the most handcrafted task that no longer resists classical simulation is surpassed by the power of artificial intelligence, which deals well with discovering relations inherent in earthquake waveform data sets. In this sense, AI provides robust tools to detect P-waves and S-waves nowadays. By exploiting single and multi-station methods, fast and accurate characterization of earthquakes could be quickly performed after strong ground motion hits by only analyzing recorded data and detecting patterns of seismic energy propagation across the seismic networks, which offer valuable insights for earthquake characterization and classification. The application of ML goes beyond traditional statistical analysis, with applications to solve the Eikonal equation through physic-informed neural networks (PINN). PINN was also proven to be rather proficient in capturing the hypocenter of an earthquake with the initial condition and imposed constraint of traditional continuum mechanics problem. This overview has enabled us to circumscribe our field of investigation during that thesis.

Starting from this, we explore more technical aspects of time-series data. Different machine learning methods can efficiently tackle the problem of reconstruction and generation and have proved to be excellent tools for investigating seismic waveform data sets. To enhance physics-based simulations with ground motion data, it has become essential to find a way to design an adapted neural network to reconstruct, generate, and manipulate reduced-order latent representations of the data. We, therefore, explore all the unsupervised representations adapted for time series. The auto-encoder, the variations auto-encoder, the adversarial learning, and the contrastive learning appear to be the step-by-step path to achieve and understand the needs of signal translation to improve the passage from physic-based acceleration time histories, yielded by high-fidelity numerical simulations to hybrid earthquake ground motion data, realistic enough on a larger frequency band. In addition to that, we have explored the main and more recent architecture adapted for time series. We have shown that the classical convolution neural network might be ill-conditioned to capture long-range dependencies inherent to seismic waveforms. We surpass this difficulty by employing transformers, conformers, and variants. Afterward, it becomes evident that our problem is an optimization problem, which must be properly defined by seeking empirical loss functions. Because time-series are essentially different from images, and spatial correlation is different from time-causality, traditional loss functions applied to the spectrograms of time-series data may fail since no translation equivariance has been proven in the time-frequency domain for seismic data, to our knowledge. We investigate the use of appropriate empirical loss function for the time domain, tailored to consider the signal's amplitude and phase. Our investigation does not stop at that since we explore other techniques to improve the reconstruction quality (features extractor and features matching) and the generation with the family of f-divergences distances. Moreover, we adopt seismology-inspired misfit criteria (Goodness of Fit criteria by Kristeková et al. (2006) [233])as a metric for the quality of the reconstruction. This metric is based on continuous wavelet transformation and criticizes the reconstruction quality at the end of the process. A score between 0 and 10 is given for the phase envelope, and the same process is done for the amplitude. Finally, we introduce the last component before addressing the core issue: choosing an adapted benchmark data set tailored to our study. We provide an overview of various data sets designed for seismologists. Following a thorough investigation, we leverage the comprehensive data set encompassing over 19,000 hours of stored seismic data from 1989 to 2018. Among the options, namely, GEOFUN, LenBD, and STEAD, we opt for STEAD, identified as the most comprehensive data set.

Chapter 3

Signal Translation with Pix2Pix

> *"So how could one live in a world where intelligence and the perception of facts meant nothing, and where authority and tradition were everything?"*
>
> — Richard Wright, Black Boy, 1945

3.1 General Idea

3.1.1 Motivation

In the last two decades, increasingly available parallel computing architectures enabled seismologists and earthquake engineers to perform impressive numerical simulations both at regional and global scales see, for instance, [236],[237], [238], [239]. The following table provides a non-exhaustive list of regional studies 3.1.

Table 3.1: Summary of major time-domain, large-scale simulations of seismic waves in the last ten years (table after [240], [241]). FDM: Finite Different Method, FEM: Finite Element Method, SEM: Spectral Element Method, DG: Discontinuous Galerkin. "-" indicates that the desired data is unavailable in the reference. This table is not exhaustive.

					Grid Size	Size	f $_{max}$
Ref.	Method	Resources	DOFs	(m)	(km×km×km)	(Hz)	Topography
Chaljub *et al.*, 2010 [242]	FDM	6 cores	-	25/125	-	2.5	no
	SEM	32 cores	66187872	150	-	2.0	yes
	SEM	63 cores	39902676	20-900	-	3.0	yes
	DG	510 cores	-	200-5000	-	3.0	yes
Bielak *et al.*, 2010 [243]	FEM	-	251457147	var	600×300×80	0.5	no
	FDM	-	2.355 billions	200	500×250×50	0.5	no
	FDM	-	5.419 billions	100	600×300×80	0.5	no
Komatitsch *et al.*, 2010[236]	SEM	192 GPU	131000256	-	chunk of earth	0.7	no
Cui et al., 2010 [244]	FDM		1308 billions	40	810×405×85	2.0	no
Rietmann *et al.*, 2012[245]	SEM	896 GPU	22 billions	24000	Western Europe× 200	0.125	yes
Taborda *et al.*, 2013[246]	FEM	24000 cores	15.9 billions	5.5 88	180×135×32	4.0	no
Heinecke *et al.*, 2014[247]	DG	1400832 cores	96 billions	-	-	10	yes
Ichimura *et al.*, 2014[248]	FEM	294912 cores	10.7 billions	0.66	2 × 2 × 0.1	-	yes
Fu *et al.*, 2017[249]	FDM	1014000 cores	23.4 trillions	8	320×312×40	18	yes
Ichumra *et al.*, 2017[237]	FEM	1179648 cores	324 billions	0.125/64	256 × 205 × 100	-	yes
Touhami *et al.*, 2022 [241]	SEM	4000 cores	13.5 billions	35 130	44 × 44 × 63	10	no
Fujita *et al.*, 2022 [250]	FEM	294912 cores	1279 billions	125	2496 × 2496 × 1100	?	yes

More recently, to achieve accurate earthquake predictions in the 0-10 Hz frequency band and beyond, scientists have developed several end-to-end computational platforms, such as the EU

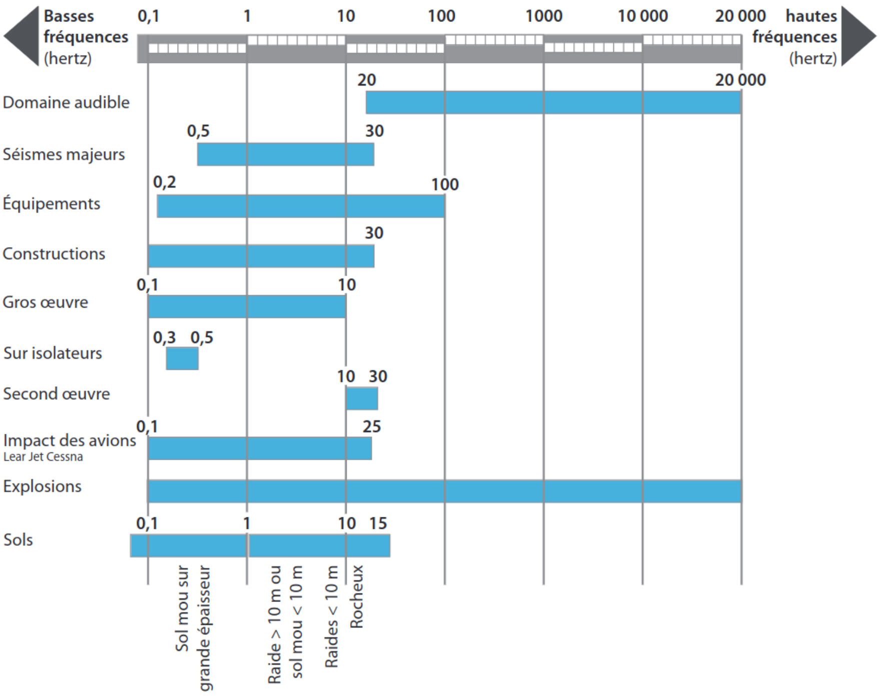


Figure 3.1: order of magnitude of natural period values (natural frequencies). ©Source Davidovici *et al.*, 2016 [20]

Center of Excellence for Exascale in Solid Earth (ChEESE) [251], EQSIM [238],[239], Cybershake [252], Cybershake-NZ [253] among many others. However, the more accurate the model is (in terms of spatio-temporal resolution of the wave propagation phenomenon reproduced numerically), the higher the associated computational cost is. In addition, even if we had infinite computing resources to reproduce an earthquake ground motion scenario in a larger frequency band, the inherent uncertainty and randomness of the natural phenomenon would remain challenging to reduce. One would need a large set of measurements characterizing the epistemic uncertainty of geological interfaces and active faults contributing to the overall seismic hazard estimation [254]. Therefore, scientific workflows and Monte Carlo simulations must be adopted see, for instance,[255],[256], to span the large dimensionality of the space of results $\mathcal{X} \subset R^d$, consisting in all the snapshots $\mathbf{u}(\mathbf{x};t)$ of any possible earthquake scenario for the region at stake. As listed in 3.1, the average largest frequency reached by regional earthquake engines ranges within the 5-10 Hz frequency band [257], [258],[259].

However, to fully characterize the dynamic transient and modal of structures and infrastructures, synthetic seismograms should be representative of the earthquake hazard level up until 20 Hz for residential buildings and up until 30-40 Hz for critical structures (For instance, see Figure 3.1), whenever adopted as free-field input ground motion for Soil-Structure Interaction studies [260],[257]. Such a brute-force deterministic approach's inherent computational burden and experimental cost are prohibitive. We can pursue a strategy with two parts : (1) reduced order and surrogate modeling of cumbersome earthquake simulations, with probabilistic modeling of the geological and seismological parameters at stake see, for instance,[261],[262],[263],[264]; (2) hybrid modeling employing numerical simulations at low frequency and empirical Green's functions [265], stochastic simulations [262], [266], or data-driven approaches (feed-forward neural networks [85], Generative adversarial Neural Networks [21, GAN],[154]) at high frequency. In particular, this work focuses on the last data-driven approach. The idea behind this hybrid broadband ground motion generation strategy is to exploit the generative power of GAN in order to decode realistic broadband seismograms from the Gaussian latent manifold. In particular, the pioneering work made by [21] first attempted to encode the physics-based low-frequency synthetic ground motion data $\mathbf{x}$ (represented by three-component acceleration traces) into a Gaussian latent space $\mathbf{z}_x = F_x(\mathbf{z}_x)$, which was next mapped into another Gaussian manifold $\mathbf{z}_y = F_y(\mathbf{z}_x)$ (via a deep generative auto-encoder) and then decoded back to the space of broadband acceleration traces $\mathbf{y} = G(F_z(\mathbf{z}_x))$ that represents a hybrid surrogate of recorded signals. Hereafter, we compare this strategy with two others issued from the well-known image super-resolution task.

The latter framework is applied to physics-based numerical simulations (analogous to low-resolution images) to generate realistic yet synthetic hybrid seismograms (the super-resolution counterpart of low-resolution images). The latter results from mapping a low-frequency signal - issued from physics-based simulation and accurate within a 0-1 Hz frequency band - into a broadband realistic signal in the 0-30 Hz frequency range. The deep learning training task is constrained by the inductive bias of the earthquake physics, integrated into the low-frequency input issued from deterministic numerical simulations. The deep learning mapping learns to render hybrid signals that preserve such an *a priori* knowledge of the earthquake scenario. The field mentioned above of research can be rephrased as a signal-to-signal translation problem. Signal-to-signal translation refers to the machine learning challenge of acquiring the relationship between an input signal and its corresponding output signal, which is another type of data. In machine learning, this task is achieved by training the statistical model on a data set containing either paired or unpaired signal instances. Initially used for solving pixel problems in computer vision and graphics, this task was later developed into image-to-image translation. In this chapter, we present the methodology employed to tackle the intricacies of signal-to-signal translation for the sake of enhancing the realism of physics-based earthquake simulations to “translate” them into broad-band time histories that can be considered as recorded seismographs of real seismic events.

3.1.2 Background on Signal-to-Signal Translation

Introducing signal-to-signal translation requires a better understanding of what has been established for image-to-image translation and audio-to-audio in machine learning before we can look into Seismology. The investigation in images is often adapted to time series and vice versa. Numerous existing studies have used Generative Adversarial Networks (GAN) for image-to-image translation, where the generation is conditioned by input data, 2018 [267]. However, these implementations often employ GAN, which unconditionally incorporates additional terms in the objective functions to constrain the output and ensure conditioning on the input. Even though this approach underscores the flexibility of GAN by allowing diverse frameworks for image translation tasks, they suffer from consistency in generations. They are not adapted for style transfer; refer to Section § 2.3.3.

In 2016, WaveNet appeared as the contribution of Aäron van den Oord *et al.*. WaveNet combines auto-regressive models and generative aspects. WaveNet was trained on more than 10000 raw audio samples to synthesize speech audio based on text. This achievement has demonstrated that text-to-audio translation and, more in general, audio-to-audio translation tasks are plausible. WaveNet key to success was Oord’s Dilated Causal Convolution layers, equivalent to masked convolution for images (see van den Oord et al., 2016). This dilated causal convolution can be implemented by constructing a mask tensor multiplied element-wise by the dilated convolution kernel. We did not find any contribution of WaveNet in the realm of seismology. Regardless, this work was the base of the audio-to-audio applications conceived by Morise *et al.*, 2016 [268], Lee *et al.*, 2019 [269] with Zhan *et al.*, 2019[270].

The pivotal contribution of Zhu in 2017, extended in 2020, presented Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, or CycleGAN [271]. This adversarial approach learns to translate an image from source domain $\mathcal{X}$ to another desired domain, $\mathcal{Y}$, without needing a paired source example. The goal of CycleGAN is to learn a mapping between these domains such that the distribution of $G : \mathbf{x} \mapsto G(\mathbf{x}) \in \mathcal{Y}$ is indistinguishable from the distribution of the targeted domain Y.

This approach has been employed in seismology by Kaur *et al.*, 2019 [272], for a seismic survey of the deep-water Gulf of Mexico. The illustration of the application of CycleGAN for interpolating seismic data is presented in Figure 3.2. Similarly, we should mention CycleGAN-VC, Which proceeds to a voice-to-voice translation, see [273]. Cai *et al.*, 2020 [274], used CycleGAN for seismic impedance inversion with semi-supervised Learning. Cia *et al.*, 2022 [275] used a variant of CycleGAN with Wasserstein distance to address surface wave tomography for the shear wave velocity inversion. Kim Dowan *et al.*, 2020 [276], used CycleGAN to overcome the imbalance problem in petrophysical facies classification. Zhon *et al.*, 2020 [277] introduced CycleGAN for time-lapse seismic reservoir monitoring data inversion. Li *et al.*, 2021 [278] also uses CycleGAN for seismic data denoising. Isola *et al.* [267], designed an Image-to-Image Translation framework with Conditional Adversarial Networks *et al.* 2018 [267] so to perform image translation. The Key part of this architecture is the use of a UNet architecture. Picetti *et al.*, 2018 [279] has proposed a generative model for seismic imaging applications based on the Pix2Pix workflow, especially for image supersampling and reflectivity deconvolution. The model has been tested on a large variability of data. Pan *et al.*, 2021 [280] used Pix2Pix as a workflow to generate 2D fluvial reservoir models that account for available field data and

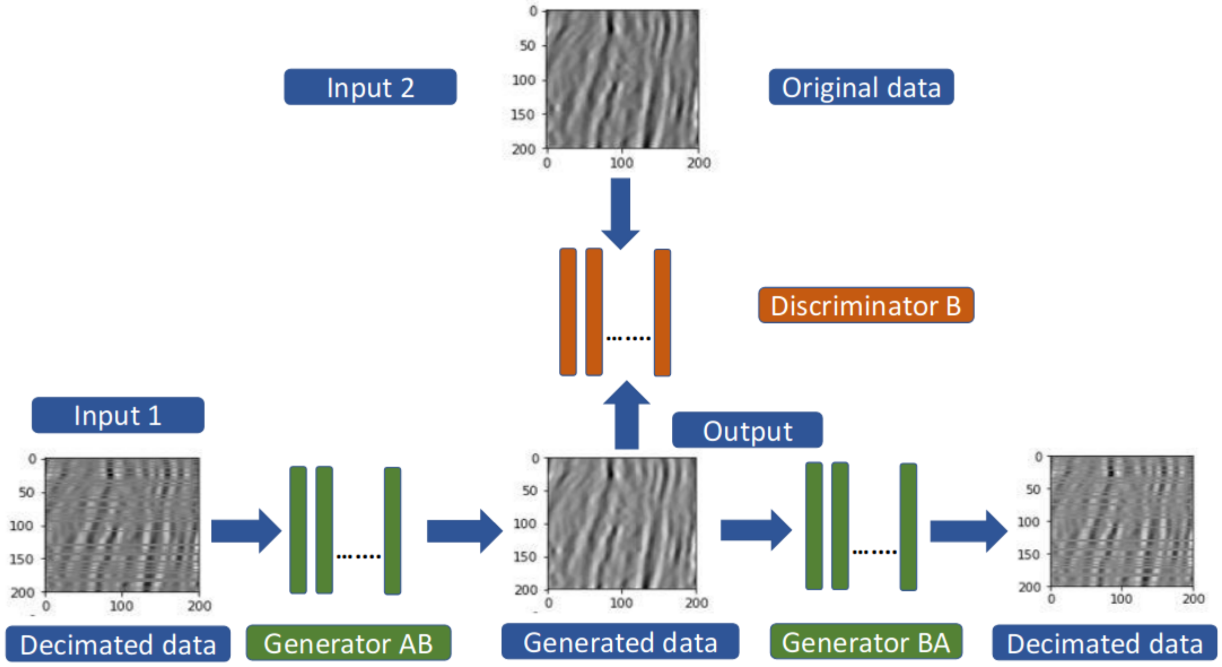


Figure 3.2: Workflow of the CycleGAN for seismic data interpolation made on 2D shot gathers data set presenting random sparsity. Source: Kaur et al., 2019 [272]

geometries of different facies. The idea of Isola *et al.*, was adapted by Sig2Sig is a transparent Signal Translation Network, by Kim *et al.*, 2021 [281], once again using UNet. Figure 3.3 shows a schematic view of Sig2Sig workflow, showcasing realistic and accurate translations. The Sig2Sig diverts from the work of Isola and his collaborator by the fact that UNet architecture is tweaked: a "squeeze and excitation" layers were added, along with multi-channel attention selection. The "squeeze and excitation" fosters better compression. In contrast, multi-channel attention selection consists of different self-attention mechanisms for generating sophisticated images and obtaining uncertainty maps for calculating loss with high weights at important pixels and low weights at noise pixels. The noisy background is, therefore, reduced by adopting the attention-selection mechanism.

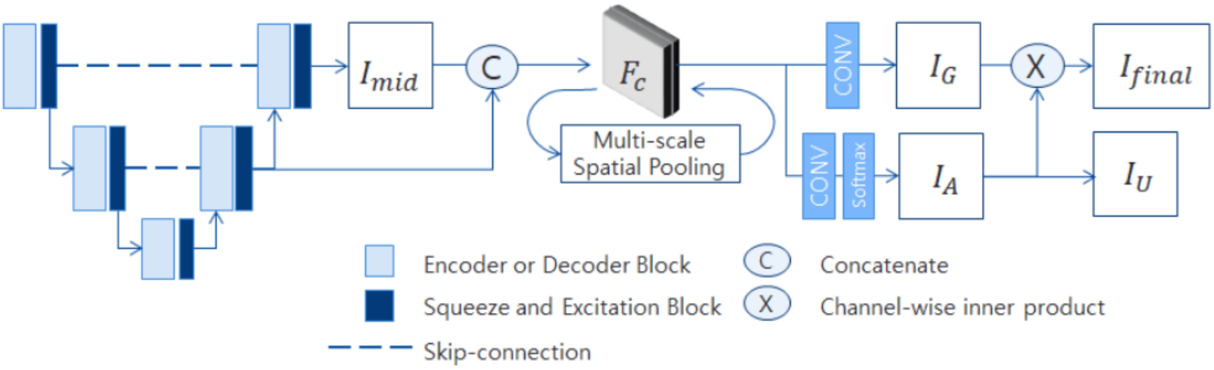


Figure 3.3: Sig2Sig framework illustrated by the work of Kim *et al.*, 2021 [281]

The Speech Enhancement Generative Adversarial Network named SEGAN [282] resembles Pix2Pix, with a UNet architecture dedicated to speech generation. Other techniques have been previously adopted in the literature for the translation task but have yet to employ in Seismology, such as DualGAN by Yi et al., 2018 [283], DiscoGAN by Taeksoo Kim *et al.*, 2017 [284], StyleGAN, by Karras *et al.*, 2020 [285], AttnGAN, by Xu *et al.*, 2017 [286], DeepSpeech2 Amodei *et al.*, 2015 [287], StyleCLIP [288].

In our data set, the term $\mathbf{x}$ represents the physic-based data in the term $\mathbf{y}$ is related to waveforms. Our strategies propose a different type of application based on the GAN workflow with Pix2Pix architecture. After observing the feasibility of the method described in the paragraph above, we intend to find a case of application to enhance the physics-based simulations without labeling and class control. Therefore, we aim to transform the simulation into recorded data from the same station. The natural uncertainty of the recorded data makes it impossible to find a unique solution for this ill-posed problem. The mapping between low and high-frequency parts of the Fourier's spectrum has never been extensively studied in seismology. Only some empirical correlations exist between the two portions of the spectrum, but to our knowledge, there is no convincing physical explanation. Therefore, Pix2Pix signal translation represents our first naive approach to solving this super-resolution problem by training a neural network to mimic this mapping based on the available training dataset.

3.1.3 Overview of Pix2Pix and its applicability

Transferring physics-based simulation to recorded ground motion data implies capturing the arrival times, the phase, and the envelope of the seismic wave. To effectively translate languages, creating a method involving paired data sets is meaningful. Therefore, inspired by language translation, the Pix2Pix architecture design can be exploited for times series. The translation aspect is treated in the subsections.

3.2 Methodology

3.2.1 Objective functions

The general idea of using Pix2Pix is to make the distribution of generated data $\mathbf{y} \sim \mathbb{Q}$ match the unknown data prior distribution, namely $\mathbf{x} \sim \mathbb{P}$, by minimizing the distance between these two probability distributions. The work of Goodfellow *et al.*, 2014 [80] proposes a way to achieve this goal through GAN. More details can be found in Section 2.3.3. Nowozin et al. [232] proposed different metrics to be used, gathered in two families: **Integral Probability Metric (IPM)** family function, as Wasserstein distance [289], [290], and f-divergence function as, Pearson $\mathcal{X}^2$ and reverse D_{KL}, see further details in Section 2.5.5. Because of his stability, the Jensen-Shannon distance has proved his ability to match distributions. This general expression is :

$$V(D,G) := \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\log(1-D(\boldsymbol{x}))]}_{\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]} \tag{3.1}$$

We define by $\mathbf{x}$ the physic-based data, by $\mathbf{y}$ the recorded data, and by $\mathbf{z}$ a Gaussian noise of 0-mean and standard deviation of 1. The functions D and G, respectively, are Discriminator and Generator. Previous work argued on the appropriate way to define D and G; see Radford *et al.*, [291]. We build a neural network with multiple bricks of a succession of Convolution, normalization output (Batch-Normalization, Instance-Normalization), activation function (LeakyReLU, ReLU), and Dropout. We will discuss the architecture later (see Section 3.2.3). Our strategy is based on the previous work of the paper of Isola *et al.*, 2018 [267], developed for image-to-image Translation, a variant of the CycleGAN [292] and conditional GAN [293]. Pair of distribution of physics-based, seismic data, and physics-based data and generated seismic data. The objective function to be satisfied is the same adopted in the original work on Pix2Pix [267], namely:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\log D(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1-D(\mathbf{x},G(\mathbf{x}))] \tag{3.2}$$

To ensure the consistency of the reconstruction term or to force the generated signal to be "close" to the original one, we adopted a Hyper-Spherical Prototype Loss(HSL) and a Focal Frequency Loss (see subsection 2.37) penalty, namely:

$$\mathcal{L}_{FFL}(G) = \mathrm{FFL}(\mathbf{y},G(\mathbf{x})) \tag{3.3}$$

$$\mathcal{L}_{HSL}(G) = \mathrm{HSL}(\mathbf{y},G(\mathbf{x})) \tag{3.4}$$

Finally, the loss to be minimized reads:

$$V(D,G) = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda(\mathcal{L}_{FFL}(G) + \mathcal{L}_{\mathrm{HSL}}) \tag{3.5}$$

In the following, the training algorithm is presented.

3.2.2 Application in case of signal translation

Pix2Pix was not designed for signal translation but for images, tested extensively on edge-to-shoes, edges-to-handbag, and day-to-night datasets (an illustration of the type of data is presented in the Figure 3.4). However, in earthquake engineering, this strategy can be used for data augmentation, as explained in the paper of Li *et al.*, 2020 [294].

Algorithm 8 The Signal-to-Signal Translation algorithm, called `Pix2Pix`

```
Intializatio of network paramters $\theta_d$, $\theta_g$
for $e$ in number of epochs do
    for $k$ in number of discriminator training step do
        Sample minibatch of $m$ raw samples $\{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(m)}\}$ from prior $p_{\text{data}}(\boldsymbol{y})$
        Sample minibatch of $m$ PBS data $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data $p_{\text{data}}(\boldsymbol{x})$
        Update the discriminator by ascending its stochastic gradient:
        $\eta_d \leftarrow \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(D\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right)\right) + \log\left(1 - D\left(\boldsymbol{x}^{i}, G(\boldsymbol{x}^{(i)})\right)\right)\right]$
        $\theta_d \leftarrow \theta_d - \eta_d$
    end for
    Sample minibatch of $m$ raw samples $\{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(m)}\}$ from prior $p_{\text{data}}(\boldsymbol{y})$
    Sample minibatch of $m$ raw samples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from prior $p_{\text{data}}(\boldsymbol{x})$
    Update the generator by ascending its stochastic gradient:
    $\eta_g \leftarrow \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(D\left(\boldsymbol{x}^{(i)}, G(\boldsymbol{y}^{(i)})\right)\right)\right]$
    $\eta_g \leftarrow \eta_g + \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \text{FFL}(\boldsymbol{y}^{(i)}, G(\boldsymbol{x}^{(i)}))$
    $\theta_g \leftarrow \theta_g - \eta_g$
end for
The Adam algorithm is commonly used for that task
```

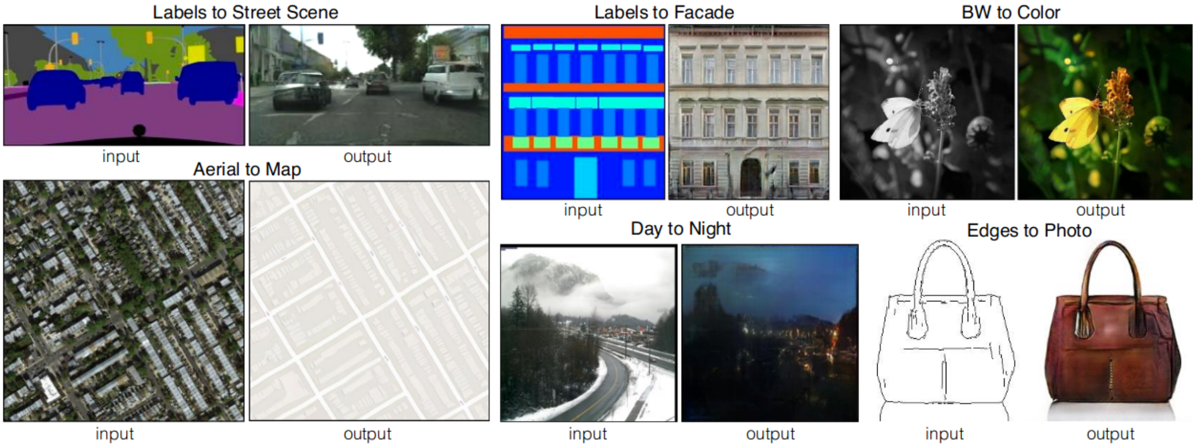


Figure 3.4: ©[267]. Isola and collaborators have developed an algorithm to leverage the process of translating image information by training models in pairs of information.

3.2.3 Pix2Pix architecture

Using an L2 loss, ℓ^2, it is possible to map physics-based simulation to raw seismic data. However, this approach does not guarantee that the Generator will adequately learn the distribution; see Section § 2.3.3. This potential limit could be circumvented by including another agent in the training, a Discriminator. Therefore, Pix2Pix architecture will involve a Generator composed of an Encoder, Decoder, and discriminator.

Generator

The chosen architecture of the Generator is a U-Net architecture. Like the Auto-Encoder, this network comprises down-sampling and up-sampling layers. A succession of convolution and convolution-transposed constitute the building blocks of such neural network (see Figure 3.7). Since Radford *et al.*, 2016 provided several principles for designing Convolutional

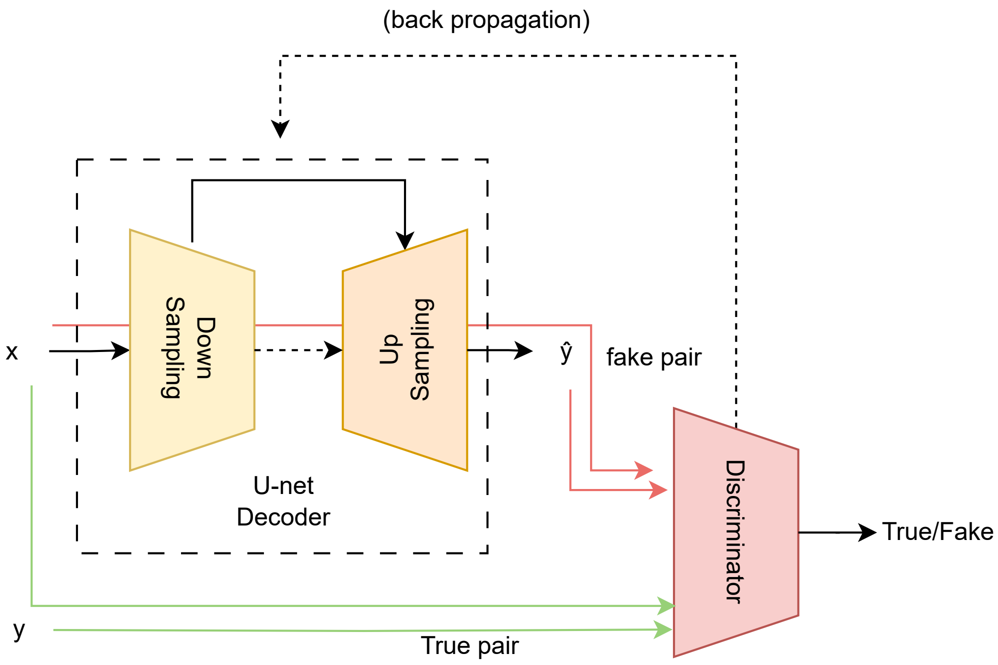


Figure 3.5: Pix2Pix model for signal translation transforms physic-based data into broadband samples. $\hat{\mathbf{y}} = G(\mathbf{x})$.

neural networks, we have created two blocks inspired by these authors: Convolution-Blocks and Convolution-Transposed-Blocks. A convolution block is in this other Convolution Layer, followed by Batch normalization and Activation function, which is LeakyReLu activation. Similarly, we use the Convolution-Transpose block with the Convolution-Transpose neuron, BatchNorm, and activation function, ReLU. We call the downsampling part the encoder, formed with convolution blocks, compressing the data into a reduced-order manifold.
In contrast, we name the upsampling part the decoder, formed of Convolution-Transposed-Blocks, generating new data samples from latent features. In addition to that, the UNet has the particularity to introduce skip-connection. The skip connection combines the corresponding level of the encoding part with the corresponding level of the decoding part. This aspect consists of concatenating the output of the Convolution block, with the upcoming output of Convolution-Transpose directly behind the inferior level of the Convolution Transpose block. Even if, in theory, we could have split the Generator architecture into two independent branches, encoder and decoder, without skip-connection joining the two, we prefer to inject the encoded features into the upsampling block through the skip-connections to condition better the generated samples translated from the original input.

We start by converting the data of shape 3×4096 to 64 ×2048, and we stop when the last convolution layers out a feature representation of 512×16; we upsample while we inject the features of the corresponding encoding level. The last value should correspond to the broadband signal of shape 3× 4096. See Table 3.2 for more details about the downsampling operation, and in Table 3.3, more details about the upsampling one. A Dropout freezes 35% of the values of the weights during the training. During the training, to avoid the model memorizing the dataset, a percentage of the weights is randomly selected so that it is not updated at every training epoch. This is a standard way to regularize neural network training since it promotes sparsity and prevents the network from learning unnecessary dependencies and over-fitting the data. In other words, dropout helps the model to generalize better. When performing the test,

Layer	**Operation**	**Input Size**	**Fonction acitavation**	**Stride**	**Output**	**Nom**
Intial Layer	Downsampling	[6, 4096]	LeakyReLU	2	[64, 2048]	D1
ConvBlock I	Downsampling	[64, 2048]	LeakyReLU	2	[128, 1024]	D2
ConvBlock II	Downsampling	[128, 1024]	LeakyReLU	2	[256,512]	D3
ConvBlock III	Downsampling	[256, 512]	LeakyReLU	2	[512, 256]	D4
ConvBlock VI	Downsampling	[512, 256]	LeakyReLU	2	[512, 128]	D5
ConvBlock V	Downsampling	[512,128]	LeakyReLU	2	[512, 64]	D6
ConvBlock VI	Downsampling	[512, 64]	LeakyReLU	2	[512,32]	D7
Bottleneck	Downsampling	[512,32]	LeakyReLU	2	[512,16]	B

Table 3.2: We present here the downsampling part for the UNet architecture. We have chosen to show the channel C and length L of the signal input and output size [C,L]. The Initial Layer is formed of the convolution layer and activation function. the Convolution Block definiton could be found in §3.2.3. A stride of 2 reduces the L dimension; we also choose to augment the number of channels; meanwhile, we divide the length of the signal by two, until a certain depth for computational cost and efficiency.

Layer	**connect. with**	**Operation**	**Input**	**Activation**	**Stride**	**Output**	**Name**
Trans.Block	’–	Upsampling	B	ReLU	2	[512,32]	UP1
Trans.Block I	D7	Upsampling	UP1	ReLU	2	[512, 64]	UP2
Trans.Block II	D6	Upsampling	UP2	ReLU	2	[512, 128]	UP3
Trans.Block III	D5	Upsampling	UP3	ReLU	2	[512, 256]	UP4
Trans.Block IV	D4	Upsampling	UP4	ReLU	2	[256, 512]	UP5
Trans.Block V	D3	Upsampling	UP5	ReLU	2	[128, 1024]	UP6
Trans.Block VI	D2	Upsampling	UP6	ReLU	2	[64, 2048]	UP7
Final	D1	Upsampling	UP7	Tanh	2	[3, 4096]	Y

Table 3.3: We present here the Upsampling part for the UNet architecture the Trans-Block performs the convolution transpose, with the bottleneck value, B, to output UP1, but does not have skip-connection. In the other step, the UP1, with the penultimate value of the Downsampling part D7 produces UP2. In the same vein, we combine UP2 with D6 to output UP2. We repeat the process until UP7. The final activation function is Tanh because database values are in [-1, +1].

the Dropout is deactivated (see Hinton *et al.*, 2012 [295]). Activation functions for compression are a LeakyReLU with a negative slope of 0.2, and we conversely use ReLU for the decoding branch. The choice of such activation functions prevents the vanishing gradient. The outputs of each Convolution and Convolution-Transposed layer are normalized over the instance with a standard Instance Norm. The Instance Normalization was defined for Style Transfer, according to Ulyanov *et al.*, 2017 [296]. However, this layer has gained a particular interest in machine learning for its performance compared to normalization per batch (BatchNorm) [296]. The definition of Instance Norm reads:

$$y_{tij} = \frac{x_{tij} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}}$$

formula in which the μ_{ti} is the mean, σ_{ti} the standard deviation of the signal.

$$\mu_{ti} = \frac{1}{L}\sum_{l=1}^{L} x_{til}$$

$$\sigma_{ti}^2 = \frac{1}{L}\sum_{l=1}^{L}(x_{til} - \mu_{ti})^2$$

A comparison of different type of normalization techniques is presented in [297], as shown in Figure 3.6.

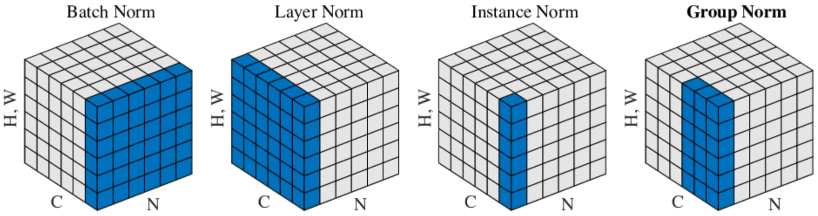


Figure 3.6: Different types of Normalization techniques present in Machine Learning. ©[297]

The reader can refer to Figure 3.7 for a visual representation of the U-Net architecture of the Generator.

Discriminator

In our design choice, we adhere to the conceptual framework of PatchGAN, according to Chang *et al.*, 2019 [298]. It is a discriminator architecture commonly employed in generative adversarial networks (GANs) for image-related tasks. Unlike traditional discriminators that produce a single output for the entire input, a PatchGAN classifies the input in a patch-wise way. See Figure 3.8 for the architecture of the Discriminator. Table 3.4 presents more details of the shape. The network then extracts relevant features. The last layer will not be a value for each sample but a 3D output feature map, to which we apply an element-wise sigmoid. Each pixel on the output map corresponds to a region of the input image to the discriminator. This means it evaluates and classifies small, overlapping patches of the input image separately. This localized assessment allows the PatchGAN to provide fine-grained feedback on the realism of different regions within the image. Designing a discriminator for images is comparatively more straightforward than for signals. In the case of audio, discerning a fake signal from an original signal poses additional complexity, as mentioned by Oord et al., 2006.[213]. Signal artifacts appear in the frequency domain. However, this latter could hardly be seen in the time domain because it is blurred by noise.

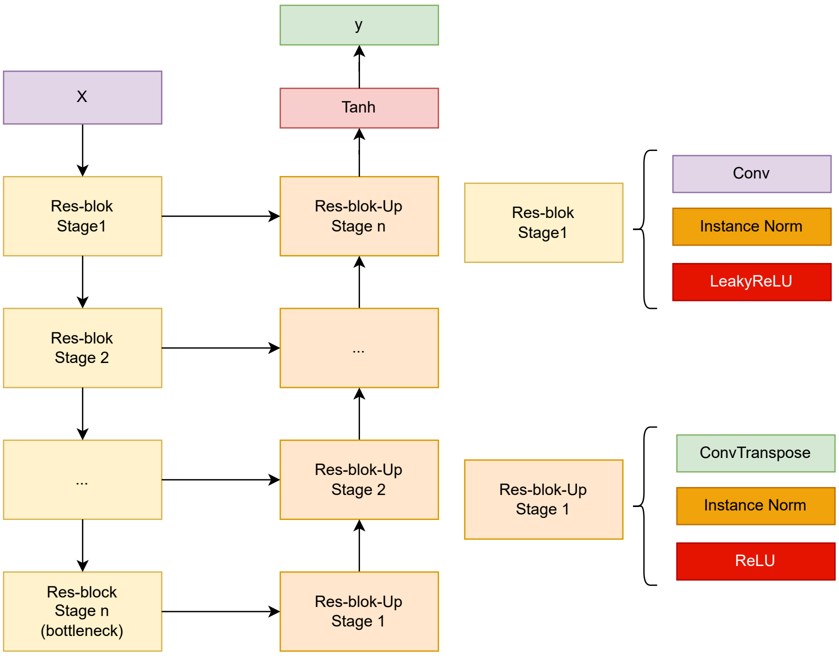


Figure 3.7: Representation of the architecture of the U-Net adopted as Generator.

Layer	**Opeation**	**Input Size**	**Fonction Activation**	**Stride**	**Output**
Initial	Downsampling	[9, 4096]	LeakyReLU	2	[16, 2048]
PatchBlock I	Downsampling	[16, 2048]	LeakyReLU	2	[32, 1024]
PatchBlock II	Downsampling	[32, 1024]	LeakyReLU	1	[64, 1023]
Final	Downsampling	[64, 1024]	LeakyReLU	1	[1, 1022]

Table 3.4: PatchCNN for the Discriminator

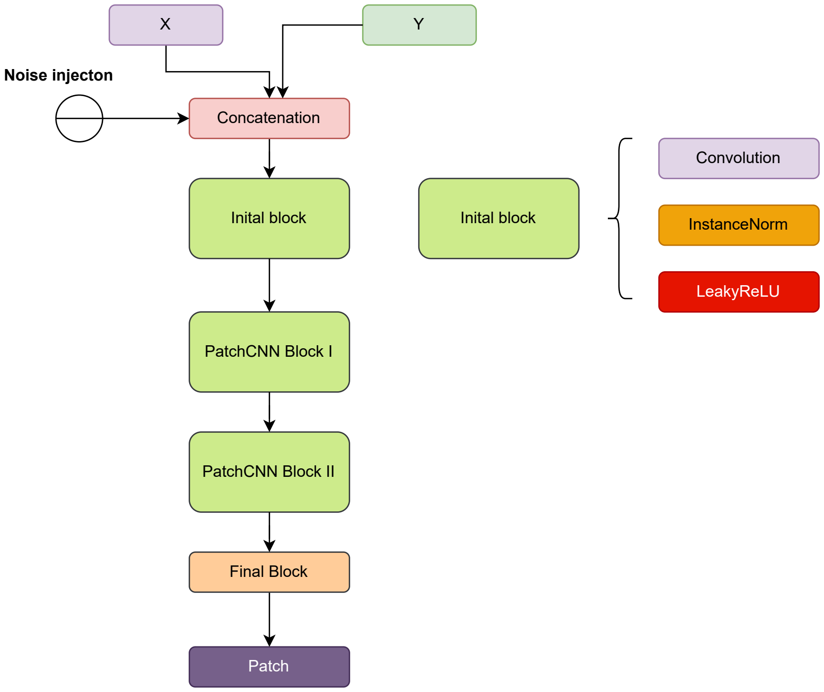


Figure 3.8: Architecture of the Discriminator adopted in the PatchGAN framework and featured by CNN layers.

3.2.4 Training Data and Dataset Description

We train our architectures by using the open source STanford Earthquake Dataset (STEAD) [235] (More details are in Section §2.7). The STEAD is a vast database that contains recorded data and metadata around the Earth. Each seismic datum is a 3-channel recorded signal in three directions. For computational power, the time history is cropped to 4096 values of the time step of 10 ms. This latter is de-trended, low-pass filtered at 30 Hz, and divided by its PGA as normalization. We perform this kind of normalization since the network could have convergence issues, and also, in a confirmed case of test, we should get access to the PGA of the signal. In machine learning, the scientist often needs thousands of data to validate a model, which are hard to obtain from high-fidelity numerical modeling because of the associated computational cost.

Moreover, given the significant uncertainty of the input parameters, it is intricate to produce an earthquake simulation that perfectly matches a seismic record. Therefore, to train our neural networks, we assumed that the recorded signals belonging to the STEAD database and band-pass filtered in a frequency range from [0-1] Hz could generate the labeled dataset required for the supervised scheme we adopted: each and any band-pass filtered signal has its target counterpart, consisting in the original broad-band STEAD record. We use 128000 3-component signals for 384000 time-histories of 4096 time-steps each. Our training set constitutes 80% of the total. We spared 10 % for validating the hyper-parameters adopted (learning rates and schedulers, penalty coefficients, etc.). The remaining 10% is used for testing the network against possible overfitting.

3.3 Experimental setup

3.3.1 Description of Evaluation Metrics

The Goodness of fit is presented in the section 2.6. The wavelet transform of a signal is used for the comparisons, in which the phase and amplitude envelopes are evaluated to determine the quality of the prediction. The generated value is $G(\mathbf{x}) = \hat{\mathbf{y}}$; this latter is compared to the original value $\mathbf{y}$.

3.3.2 Implementation Details

Before completing the training, we do a unitary test on parts of our architecture. On the one hand, we tested the U-Net architecture with a ℓ^1- and ℓ^2-norm for reconstruction and Focal Frequency Loss. Indeed, this simple loss evaluation could not be robust enough to capture high-frequency information from our data set. However, this lets us conclude whether our architecture has sufficiently generalized to map two data sets. It helps us understand that adding noise to the Generator does not help improve the model. The network will learn to ignore this latter, as proven [267]. On the other hand, the evaluation of discriminators is made separately. We randomly generate Gaussian noise for this part, concatenated with the original signal and uniform distribution from 0 to 1, taken as artificial data. We start with the Gaussian noise with a standard deviation 1.0 and progressively decrease the standard deviation until 0.10. Finally, the same original data replaces this artificial data with noise. We use the binary cross entropy loss for this task. After calibration and separately validating unitary tests, we bind the blocks and train the model. This strategy saves time finding which part of our architecture must be fixed. Adding noise to this strategy is relevant to convergence. Noise addition makes generalization possible and avoids some artifacts in the generation's output.

Technical detail

We trained Pix2Pix with the distributed data-parallel scheme on 4 NVIDIA P100 GPUs and 4 NVIDIA A100 GPUs at the Mesocentre Moulon supercomputer facility at Université Paris Saclay. The previous work of Gatti *et al.*, 2020[21] made in the Python version of PyTorch (v1.8, CUDA 10.2), has served as the base of the experiment. This version has been optimized to version 2.0 of Python and CUDA 11.7.

Parameter Tuning and Optimization

One notable challenge in machine learning, particularly in GAN and its variants, is their high sensitivity to hyper-parameters, Dumont *et al.*, 2022 [299]. Minor variations in architecture, optimization techniques (Adam, RMSprop, SGD, etc.), learning rates, or use of inappropriate schedulers can profoundly impact prediction or generation quality. Further details on specific challenges are deferred to the experimental section below. All the state-of-the-art GANs and associated regularization methods that serve as guidelines in our studies can be found in Table 5 of Lee *et al.*, 2020 [165]. The choice of the Adam optimizer was based on its stability, ensuring uninterrupted training even as gradients approach zero ($\beta_1 = 0.5$ and $\beta_2 = 0.999$). Addressing the challenge of selecting appropriate learning rates for the Generator and discriminator involved employing the Two Time Update Rule (TTUR) [300]. Specifically, a learning rate of 0.0004 was applied to the Generator and 0.00001 to the discriminator, with the latter being 10 times smaller than the former. A MultiplicativeLR, Dong *et al.*, 2022 [301] scheduler was adopted, which reduces the learning rate values slightly during the training. This technique helps stabilize the training and avoid the loss of jumps up and down. More than that, we apply a weight decay of 0.0001 to the discriminator; by adjusting the weight decay coefficient, we can control the balance between fitting the training data and preventing overfitting.

Following an exhaustive search and tuning process, optimal hyper-parameters were determined. A U-Net architecture was adopted since it shows compelling convergence throughout the training process, effectively minimizing prediction errors.

3.4 Results and Discussion

After completing the training phase, we applied our U-Net architecture to the STEAD dataset. Our findings demonstrate that the physics-based data $\mathbf{x}$ transforms $\hat{\mathbf{y}}$ when compared to the paired data. Figure 3.9 illustrates the network's capability to reproduce seismographs in the 3-directions (North-South, East-West, Up-Down). The Goodness of Fit provides us a metric reflecting the quality of Translation, as shown in Figure 3.10 and Figure 3.11, for the test dataset.

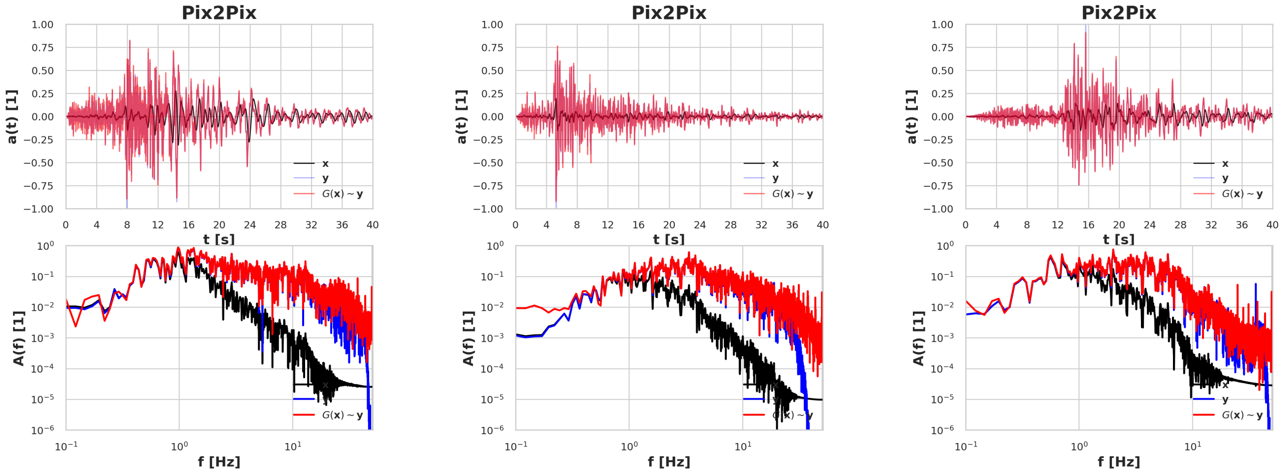
Reminder

As a reminder, as described in subsection 2.6), the score proposed by Kristeková *et al.*, 2009 [234], is given for both the phase and amplitude on a 0-10 scale. A score of 10 expresses a perfect reconstruction quality.

In earthquake engineering, notable approaches for hybrid generation include the prior works of Paolucci *et al.*, 2018 [302], and Gatti *et al.*, 2020 [21].

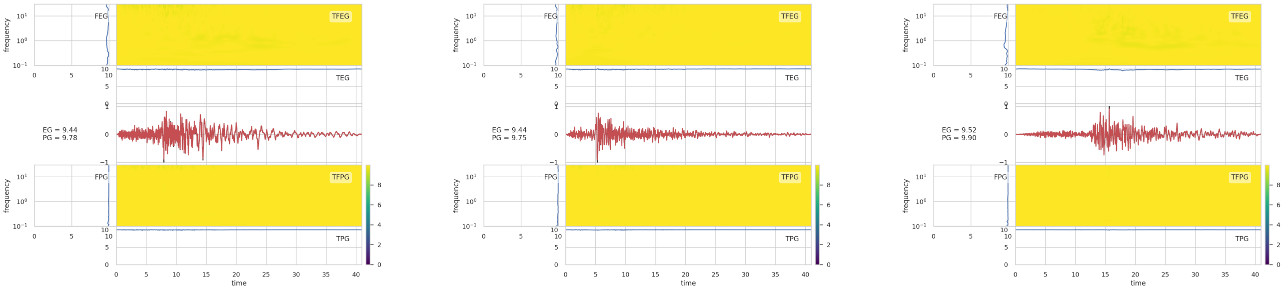
Paolucci *et al.*, 2018 addressed broadband ground motion generation utilizing their architecture named ANN2BB, which is based on standard multi-layer perceptron to predict Pseudo-Spectral Acceleration (PSA, damped at 5%) at short periods, specifically targeting response spectrum values for natural periods lower than 0.75 s, see Figure 3.12

These short-period values are crucial for engineering design purposes but are often challenging to obtain by physics-based numerical simulations. Those short-period PSA ordinates can be reliably computed if the numerical simulation that renders synthetic ground motion time-histories is sufficiently accurate at high-frequency, *i.e.*, in a 0-30 Hz frequency range.



(a) North-South (b) East-West (c) Up-Down

Figure 3.9: Earthquake signal translation with Pix2Pix. The signal in blue, **y** is the target signal. It is a broadband signal with frequency values between 0 and 30 Hz. The signal in black, **x**, mimics the numerical simulation (in fact, it is a filtered version of the broadband signal for frequencies around 0 and 1 Hz.). The U-Net architecture, trained in Pix2Pix fashion, can provide a good reconstruction in red that is close to the broadband signal.



(a) North-South (b) East-West (c) Up-Down

Figure 3.10: We present the G.O.F, which evaluates the signal in amplitude and phases. A score of 0 is the worst, and 10 is the best reconstruction. As we can see, by training pix2pix using the U-Net strategy, we can have an accurate reconstruction of **y** from **x**

ANN2BB renders realistic PSA values below T =0.75 s, based on the longer-period ones (T >0.75 s), assumed to be accurately evaluated by simply adopting the physics-based simulation **x**, valid in a 0-1.5 Hz frequency band. The predicted PSA, as shown in Figure 3.12, is adopted as a target to iterative scaling the Fourier spectral amplitudes $|\mathcal{F}(\mathbf{x})(\omega)|$ (with ω being the natural pulsation and $\mathcal{F}(\cdot)$ the Fourier transform operator) of the original physics-based signal **x** until its PSA values $Sa(T)$ aligns with the target ANN2BB target one. An illustration summarizing the methodology can be found in Figure 3.12.

Paolucci's ANN2BB method involves several steps before obtaining the desired generation results. Additionally, the chosen architecture of the Artificial Neural Network (ANN) may struggle to capture long-range dependencies. Furthermore, it is noted that the procedure is not yet suitable for obtaining sets of waveforms with realistic spatial coherency features at high frequency. Addressing these limitations, Gatti and Clouteau [21] proposed an alternative approach. Rather than focusing on response spectral ordinates, they advocate working with the entire time series to generate hybrid signals. However, this approach has its own set of limita-

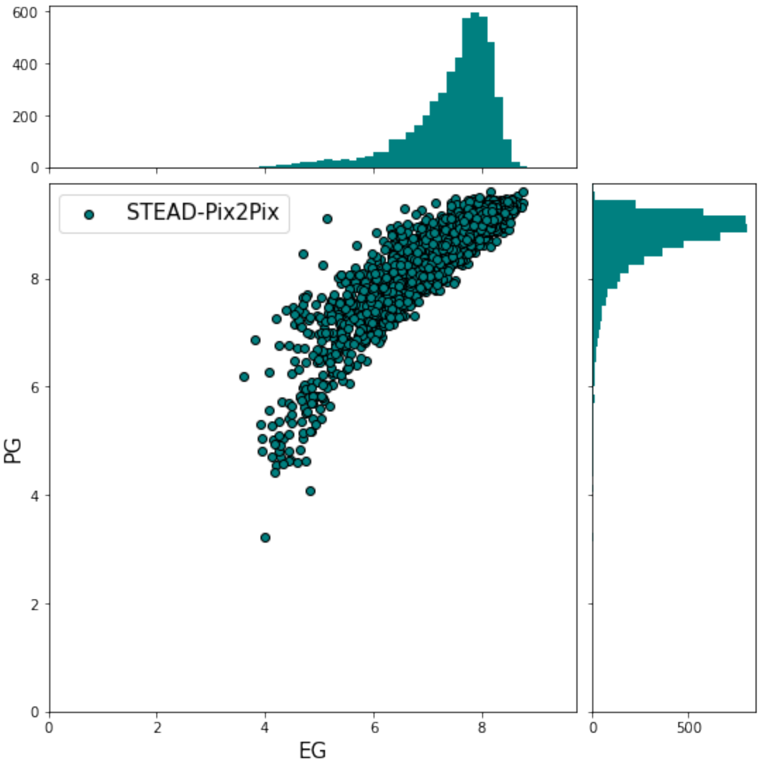


Figure 3.11: Goodness of Fit (GoF) metrics values are obtained for Pix2Pix to evaluate the quality of hybrid signal generation from physics-based data. The generated signals should closely match the targeted ground motion data.

tions, which will be discussed in the following chapter. One noteworthy limitation is achieving high-quality reconstruction of hybrid generation in terms of fidelity to the targeted values. It appears clear that traditional methods, as exemplified by Paolucci *et al.*, 2018, and Gatti and Clouteau, 2020, involve multi-step processes and architectural considerations that might need help capturing long-range dependencies in the data. These dependencies are crucial in seismic signals because they contain information distributed over a wide range of time series. Pix2Pix, on the other hand, simplifies the process. We do this by minimizing the joint distribution. In addition, we examine and integrate features associated with long-range dependencies using the U-net architecture. Our methodology aims to capture complex correlations in seismic data in a single step. This makes it possible to generate more efficient and accurate hybrid signals. Finally, our strategy is designed to enhance the process's computational efficiency and improve the overall fidelity of the generated hybrid signals by directly addressing the complexities associated with long-range dependencies in seismic data. The output of our time-domain results is illustrated in Figure 3.9 for a seismograph, showcasing outputs in three directions to demonstrate both time and frequency domain reconstructions. In the time domain, the generated acceleration exhibits the characteristic appearance of a seismic signal. The signal's amplitude initiates from zero, exponentially increasing and subsequently decreasing, ensuring the generated signals' plausibility. The potency of matching joint distributions, specifically $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}, G(\mathbf{x}))$, is evident in achieving this realistic seismic signal generation.

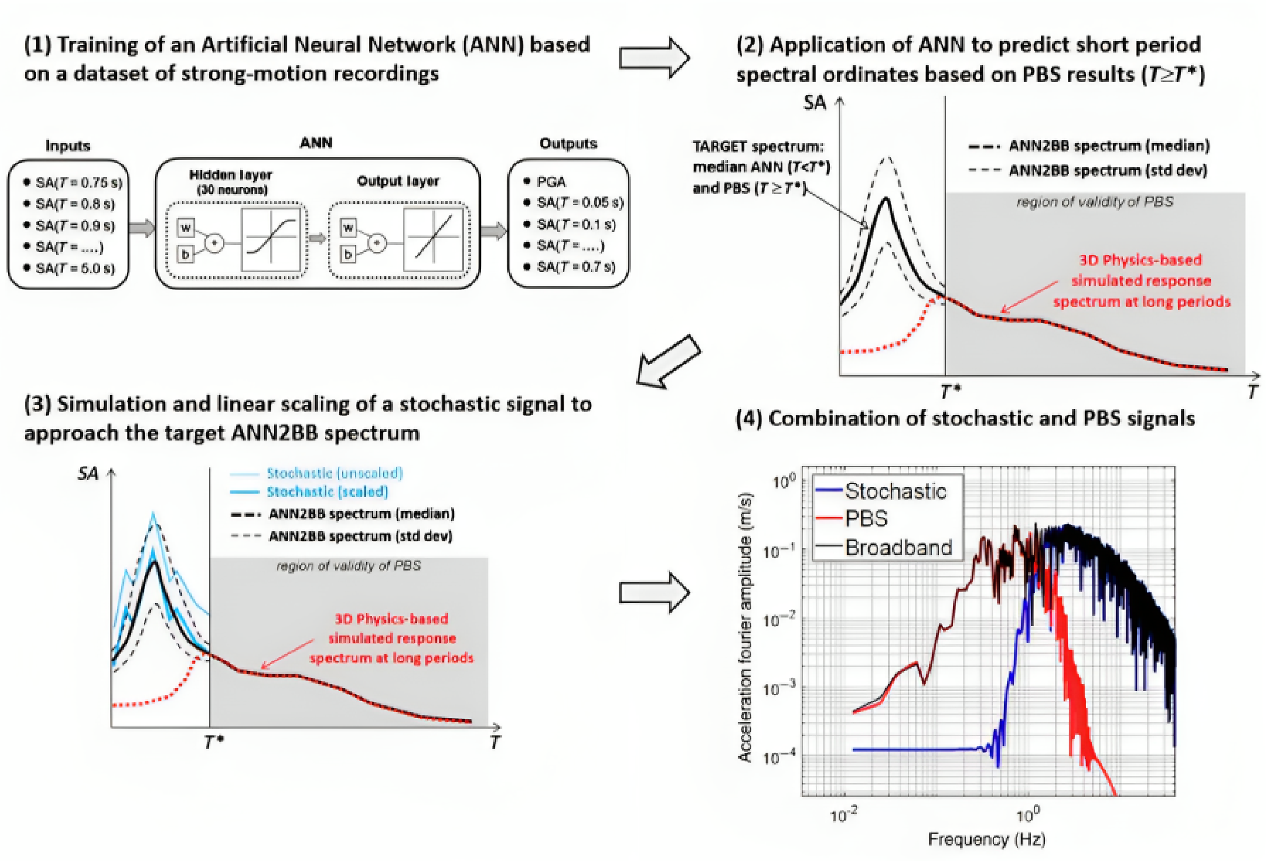


Figure 3.12: Flowchart of the ANN2BB approach revised after Paolucci et al. (2018) for the massive processing of physics-based numerical simulations (PBS) for broadband computation. Source [303]

Reminder

PSA values, namely the application $T \in \mathbb{R}^+ \mapsto Sa(T) \in \mathbb{R}^+$ corresponds to the maximum absolute value of the oscillator response $(t, T) \mapsto y_T(t)$, with natural period T, to the earthquake acceleration $t \in [0, \Delta] \subset \mathbb{R}^+ \mapsto \ddot{x}_g(t)$, solution of the following ordinary differential equation:

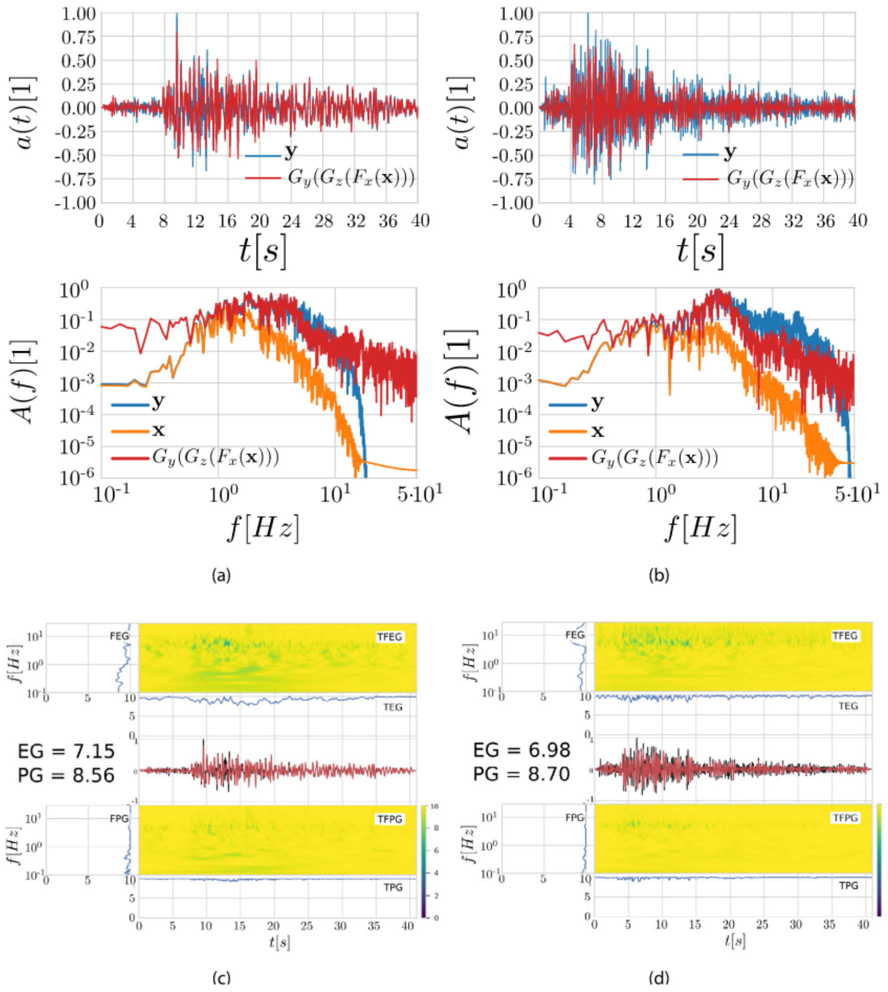
$$\ddot{y}_T(t) + \frac{4\pi\zeta}{T}\dot{y}_T(t) + \frac{4\pi^2}{T^2}y_T(t) = -\ddot{x}_g(t)$$

with ζ the critical damping, $\Delta > 0$ the earthquake duration and the Pseudo-Spectral Acceleration computed as:

$$Sa(T) = \max_{t\in[0,\Delta]} |y_T(t)|$$

In earthquake engineering, the PSA is widely adopted to assess the approximate seismic response of buildings underground shaking, considering such structures as single-degree-of-freedom oscillators, whose natural oscillation period approximately corresponds to $T \approx C_t \cdot h_n^x$, with h_n being the structural height, and C_t, x some coefficients depending on the type of building (reinforced concrete, steel frames, masonry, etc., according to the ASCE code 7-16 Section 12.8.2.1 Approximate Fundamental Period).

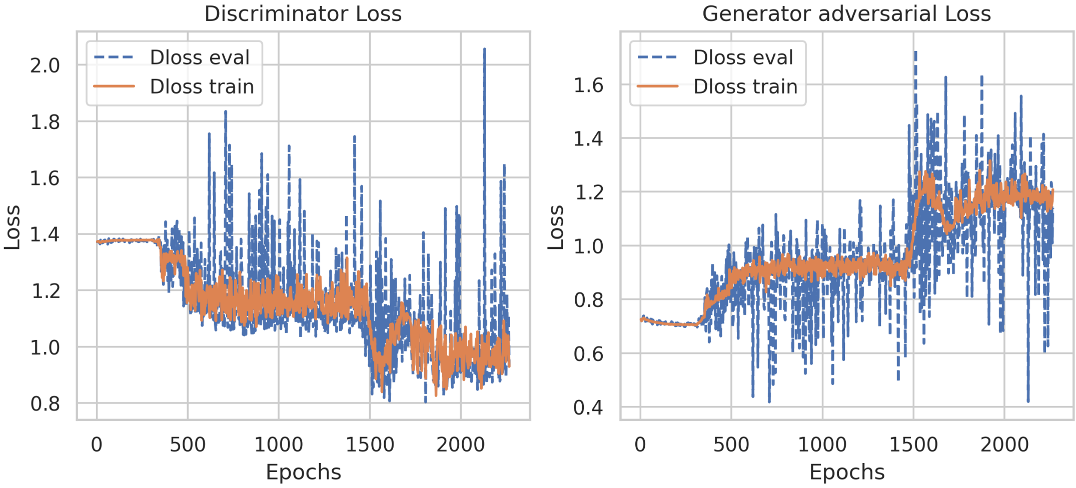
It is noteworthy that Pix2Pix learns to translate signal-to-signal in the time domain. Consequently, examining the frequency domain is just for the sake of clarity, independently of how the generators synthesize the data. In the Fourier domain, the ordinate values satisfactorily approximate the targeted values. Notably, generating values becomes challenging for frequencies between 0-1 Hz when they are lower than 10^{-3}. The Goodness-of-Fit evaluation for each direction is depicted in Figure 3.10. This analysis provides insights into the alignment between the generated signals and the target values, further assessing the performance of our approach. The Pix2Pix strategy demonstrates a comparable level of prediction quality as SeismoALICE, developed by Gatti and Clouteau [21]; refer to Figure 3.13. Figure 3.11 shows the Goodness-of-Fit (GOF) scores for the entire tested dataset of STEAD. The test dataset encompasses 12800 3D signals, totaling 38400 individual signals. The mean score for the envelope is around 8; similarly, the mean score for the phase is eight. In their seminal work, Gatti and Clouteau [21] only used 100 3-component time series for training. While most signals exhibit a high level of translation fidelity, some signals pose challenges due to their sparse ordinate values, resulting in an enhancement that deviates significantly from the targeted signals. This observation underscores the nuanced nature of signal translation, mainly when dealing with signals that inherently lack sufficient information. The evolution of the adversarial loss across the whole training epochs is shown in Figure 3.14.



(a) (b)

(c) (d)

Figure 3.13: We can see the best values of hybrid generation from SeismoALICE, developed by Gatti and Clouteau [21], $G_y(G_{(}F_x(\mathbf{x})))$ that should target the corresponding raw data, $\mathbf{y}$



(a) Adversarial Losses $\mathcal{L}_{cGAN}(G, D)$.

(b) Focal Frequency Loss $FFL(\mathbf{y}, G(\mathbf{x}))$.

Figure 3.14: Evolution of Pix2Pix training and testing losses, for signal-to-signal translation, in Equation 3.5.

3.5 Challenges and Considerations

3.5.1 Limitation of Pix2Pix in Signal-to-Signal Translation

The Pix2Pix framework exhibits notable limitations in signal-to-signal Translation that must be addressed. The Pix2Pix is primarily restricted to paired data, limiting its functionality to a type of data set exclusively: this architecture could only be trained in a supervised manner, but this condition is hard to achieve in practice because that would require a high-fidelity cumbersome numerical simulation for each earthquake and each station. This is impossible because of the large lack of information on the nature of the underground medium and active fault characteristics, which prevents the possibility of fully constraining the numerical models designed for a specific earthquake scenario. Moreover, empirical investigation of this framework has highlighted its sensitivity to noise. In the signal context, this could completely change the signal from the targeted ground motion data.

Additionally, the interoperability of the output poses a challenge; Pix2Pix prediction could be hard to understand and explain. In the case of time histories, if the inputted signal does not follow the same distribution, the output could not be expected to be realistic according to standard seismological criteria (wave arrival times, peaks, exponential envelope decay after the strong S-wave arrival). Instances of misalignment of the S-Wave and the P-Wave indicate potential discrepancies, comprising the model's fidelity to signal characteristics. Furthermore, Pix2Pix is limited to one-to-one mapping, constraining its applicability in earthquake engineering and probabilistic seismic hazard assessment, necessitating many plausible broadband realizations of the same earthquake. Therefore, the model may fail to capture the entirety of signal variation. Moreover, Pix2Pix shows its limitations in signal generation; this architecture could not be used for data augmentation, imposing constraints on its utility.

3.5.2 Overcoming challenges and Futures Directions

The nature of our dataset, which consists of paired data, ensures the alignment of phases, and we intend to maintain this approach for future architectures, presented in the following chapters. Adversarial training methods use Gaussian distributions to capture high-level features. To control the quality of the generation, it is helpful to use an encoder to project the data into Gaussian space. We will cover these details in the next chapter. The mapping onto multiple domains will involve the utilization of both PBS and Gaussian noise. The exploration of an adapted strategy should focus exclusively on this combination. We will employ a joint distribution to assess the hybrid pairs of $\mathbf{x}$ and $\mathbf{y}$ to mitigate the noise impact. Introducing a discriminator will play a crucial role in stabilizing the training process.

3.6 Conclusion

In summary, the Pix2Pix framework has demonstrated its ability to translate PBS signals into broadband data. The training was performed on the STEAD dataset, revealing the strategy's feasibility. Compared with these strategies, ANN2BB and Seismo-ALICE, Pix2Pix gives a U-Net architecture the ability to produce plausible results. However, empirical observations have revealed limitations in its ability to generate diverse hybrid signals from the same data type. In response to these challenges, we plan to address and rectify these limitations in the next chapter, which will be devoted to further methodological advances.

Chapter 4

SeismoALICE

> *"Nervous people are amazing!*
> *Whether they're estimable or not,*
> *nothing they do is banal;*
> *extravagant perhaps, astonishing,*
> *admirable, crazy, pyramidal or*
> *simply mind-boggling, that's for*
> *sure!"*
>
> — Jacques Stephen Alexis,
> L'Espace d'un cillement,1959

4.1 General Idea

As introduced in subsection 2.3.4, Li *et al.*, 2017 [175] proposed an Adversarial Learning Inference strategy with Cross Entropy, namely ALICE. The latter framework is based on the previous work of Dumoulin *et al.*, [304] 2017, called Adversarial Learning Inference (ALI). ALI and ALICE were developed to train a GAN efficiently—a schematic view of how ALI and ALICE work is presented in Figure 4.1. The original GAN strategy proposed by Goodfellow *et al.*, 2014 [80] infers data samples $G(\mathbf{z})$ from a Gaussian manifold $\mathcal{Z}$ via a neural network named Generator $G : \mathcal{Z} \rightarrow \mathcal{X}$. ALI tries to complete the task by forcing the Generator to learn the inverse mapping $\mathcal{X} \rightarrow \mathcal{Z}$, called "inference", from the data space to the Gaussian manifold. In this case, a new network architecture, the encoder G_x, is introduced, and the GAN adversarial loss is reformulated so that the joint probability distribution of data and latent space variables is considered. In other words, as in the previous classic GAN, an objective cost function should satisfy the Nash equilibrium by solving the following min-max problem:

$$\begin{aligned}
\mathcal{L}_{\mathrm{ALI}} = \min_{G_x,F_x} \max_{D_{xz}} V(D_{xz}, G_x, F_x) &= \mathbb{E}_{q(\boldsymbol{x})}\left[\log\left(D_{xz}\left(\boldsymbol{x}, F_x(\boldsymbol{x})\right)\right)\right] \\
&+ \mathbb{E}_{p(\boldsymbol{z})}\left[\log\left(1 - D_{xz}\left(G_x(\boldsymbol{z}), \boldsymbol{z}\right)\right)\right] \\
&= \iint q(\boldsymbol{x}) q(\boldsymbol{z} \mid \boldsymbol{x}) \log(D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z} \\
&+ \iint p(\boldsymbol{z}) p(\boldsymbol{x} \mid \boldsymbol{z}) \log(1 - D(\boldsymbol{x}, \boldsymbol{z})) d\boldsymbol{x} d\boldsymbol{z}
\end{aligned} \tag{4.1}$$

where two Generators G_x and F_x and a Discriminator D_{xz} are adopted, so that the probability distributions $q(\mathbf{X}, G_x(\mathbf{X}))$ and $p(F_x(\mathbf{Z}), \mathbf{Z})$ converge towards the equilibrium distribution of the

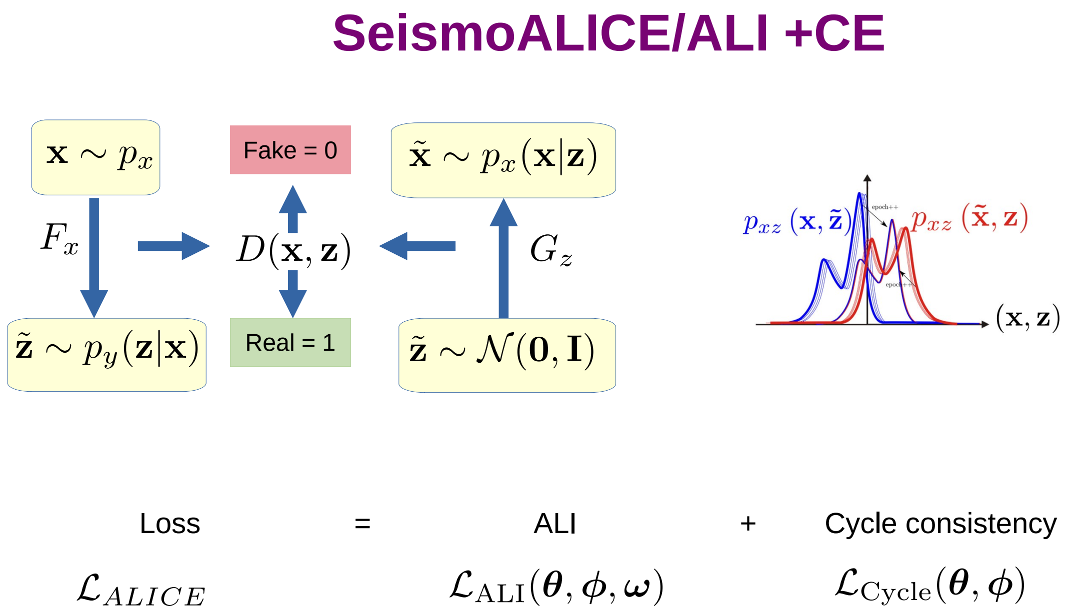


Figure 4.1: ALICE representation of the technique

mixture $\frac{q(\mathbf{X},G_x(\mathbf{X}))+p(F_x(\mathbf{Z}),\mathbf{Z})}{2}$, reaching the Nash equilibrium, *i.e.* the minimum of $V(D_{xz}, G_x, F_x)$ in Equation 4.1. As mentioned by Goodfellow *et al.*, 2014 [80], when the Discriminator takes advantage of the two Generators and gets close to 1, the two generators hardly find a way to minimize the loss functions, unless their gradient becomes increasingly large (gradient explosion). The balance between Discriminator and Generators is achieved when the Discriminator cannot distinguish samples $(\mathbf{x}, G_x(\mathbf{x}))$ from samples $(F_x(\mathbf{z}), \mathbf{z})$, since the Jensen-Shannon divergence between $q(\mathbf{X}, G_x(\mathbf{X}))$ and $p(F_x(\mathbf{Z}), \mathbf{Z})$ goes to 0. In the case of ALI, even if a couple of networks (encoder G_x, Decoder F_x) adequately describe the relationship between data and latent space representation, nothing guarantees this mapping is bijective. To fix this inconsistency, the author introduces the Cross-Entropy (CE) penalty and the original ALI loss in Equation 4.1.
However, as explained by Radford et al., 2016 [167] "*[...]the objective functions do not constraint the relationship between the latent variables and the observations, which results in unsatisfied reconstruction performance.* ALICE resolves this non-identifiability issue by optimizing the conditional entropy [175]. Cross-entropy is achieved by a couple of discriminators, who must distinguish between the original information and its perfect reconstructions. The Cycle consistency loss enforces the match of the marginal probability, *i.e.*:

$$\begin{aligned}\mathcal{L}_{\text{Cycle}} = \min_{G_x, G_z} \max_{D_{xx}} V(D_{xx}, F_x, G_x) = \\ \mathbb{E}_{q(\boldsymbol{x})}\left[\log\left(D_{xx}\left(\boldsymbol{x}, \boldsymbol{x}\right)\right)\right] + \mathbb{E}_{p(\boldsymbol{z})}\left[\log\left(1 - D_{xx}\left(\boldsymbol{x}, G_x \circ F_x(\boldsymbol{x})\right)\right)\right]\end{aligned} \tag{4.2}$$

For example, a signal and its reconstruction encoded and then decoded should not change. In addition, we enforce the mapping by an MSE loss or ℓ^1 loss multiplied by arbitrary values to harmonize the total objective function to be satisfied, see Figure 4.1

$$\mathcal{L}_{\text{ALICE}} = \mathcal{L}_{\text{ALI}} + \mathcal{L}_{\text{Cycle}} \tag{4.3}$$

4.1.1 Architecture of SeismoALICE

To apply ALICE to the super-resolution problem in earthquake engineering, Gatti and Clouteau, 2020 [21] subdivided the learning problem into three sub-problems. Two hidden variables ($\mathbf{z}$ and $\mathbf{z}'$) is introduced, for which the distribution is a mathematically well-known Gaussian or uniform. To transform the seismic signal to a physic-based simulation and vice versa, Gatti and Clouteau, 2020 [21] had to find a reduced abstract representation of each data and define a bridge between them, as shown in Figure 4.2.

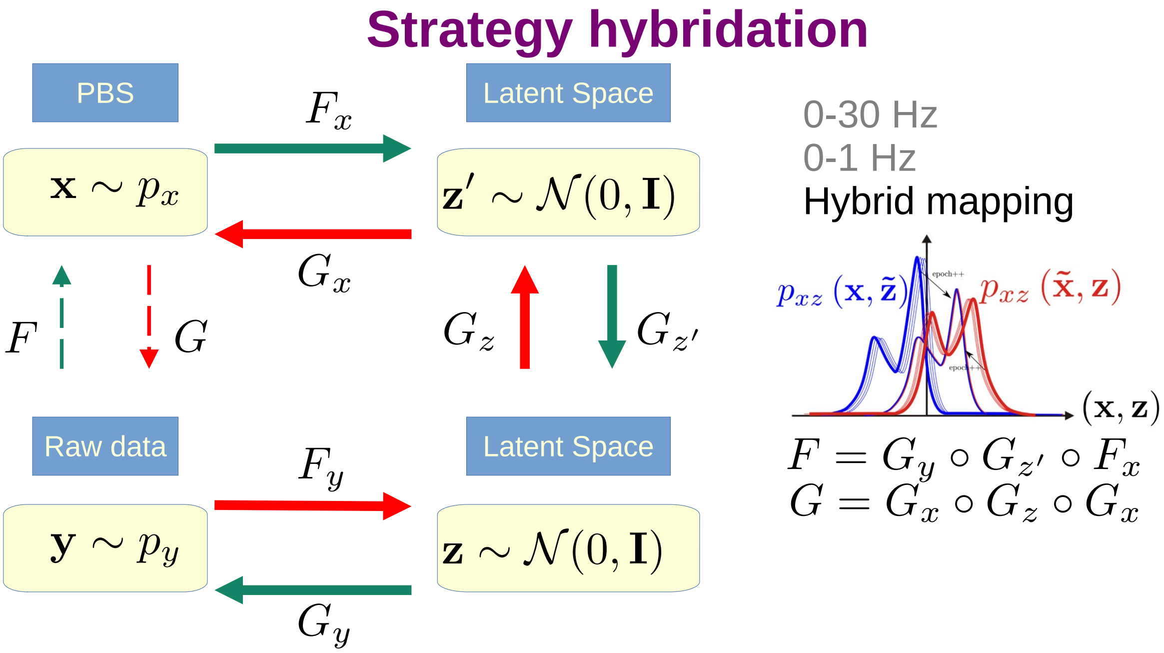


Figure 4.2: Resume of the SeismoALICE framework to passe from Physic-Based signal to ground motion.

The first problem, referred as to $\mathcal{P}_x$, consists of finding a couple of Decoder $G_x : \mathbb{R}^r \longrightarrow \mathbb{R}^d$ and encoder $F_x : \mathbb{R}^d \longrightarrow \mathbb{R}^r$ for physics-based numerical time-histories, such a way that the couple $(\mathbf{z}, G_x(\mathbf{z}))$ and $(F_x(\mathbf{x}), \mathbf{x})$ be identically distributed. As in the ALICE framework, two discriminators are adopted to drive this learning task. The second problem, $\mathcal{P}_y$, consists into finding a couple of Decoder $G_y : \mathbb{R}^r \longrightarrow \mathbb{R}^d$ and encoder $F_y : \mathbb{R}^r \longrightarrow \mathbb{R}^d$ to match the distribution of $(\mathbf{z}', G_y(\mathbf{z}'))$ and $(F_y(\mathbf{y}), \mathbf{y})$, corresponding to recorded seismograms. Finally, Gatti and Clouteau, 2020 [21] introduced the problem $\mathcal{P}_z$, in which we have to find $G_z : \mathbb{R}^r \longrightarrow \mathbb{R}^R$ and $F_z : \mathbb{R}^R \longrightarrow \mathbb{R}^r$ to map the latent variable corresponding to $\mathbf{x}$ into $\mathbf{y}$: $(\mathbf{x}, G_y \circ G_z \circ F_x(\mathbf{x}))$ and $(G_x \circ F_z \circ F_y(\mathbf{y}), \mathbf{y})$. So the composed encoder and Decoder read $F = G_x \circ F_z \circ F_x$ and $G = G_y \circ G_z \circ F_x$ respectively. All G_i and F_i mentioned above were modeled through deep CNN, as detailed in the original publication [21].

In $\mathcal{P}_x$, the cycle consistency and ℓ^1 loss were adopted so that our network could provide accurate reconstruction. The overall loss function reads:

$$\begin{aligned}
l^{x}(\mathbf{z},\mathbf{x},C,F_x,G_x) &= l_{D_{xz}}(\mathbf{z},\mathbf{x},C) + l_{D_x}(\mathbf{z},\mathbf{x},C) + l_{D_z}(\mathbf{z},\mathbf{x},C) + \lambda_x l_{R_2}(\mathbf{x}) + \lambda_z l_{R_2}(\mathbf{z}) \\
l_{D_{xz}}(\mathbf{x},F_x(\mathbf{x}),C) &= \log D_{xz}(\mathbf{x},F_x(\mathbf{x}))\mathbf{1}_{c=\text{"}d_x\text{"}} + \log(1-D_{xz}(G_x(\mathbf{z}),\mathbf{z}))\mathbf{1}_{C=\text{"}g_z\text{"}} \\
l_{D_x} &= \log D_x(\mathbf{x}) + \log\left(1 - D_x \circ G_x(\mathbf{z})\right) \\
l_{D_z} &= \log D_z(\mathbf{z}) + \log\left(1 - D_z \circ F_x(\mathbf{x})\right) \\
l_{R_2}(\mathbf{x}) &= \|\mathbf{x} - G_x \circ F_x(\mathbf{x})\|_1^2 \\
l_{R_2}(\mathbf{z}) &= \|\mathbf{z} - F_x \circ G_x(\mathbf{z})\|_1^2
\end{aligned} \tag{4.4}$$

As far as $\mathcal{P}_y$ goes, the same considerations apply, and the loss function reads:

$$\begin{aligned}
&l^{h}\left(\mathbf{z},\mathbf{z}',\mathbf{x},\mathbf{y}\right) = l^h_y(\mathbf{x},\mathbf{y}) + l^h_x(\mathbf{x},\mathbf{y}) + l^h_{z'}\left(\mathbf{x},\mathbf{z}'\right) + l^h_z(\mathbf{z},\mathbf{y}) + \\
&\lambda l_{R1}\left(\mathbf{y}, G_y \circ G_z \circ F_z \circ F_y(\mathbf{y})\right) + l_{R1}\left(\mathbf{x}, G_x \circ F_z \circ F_y(\mathbf{x})\right)
\end{aligned} \tag{4.5}$$

$\mathcal{P}_z$ aims at finding a couple of neural networks to map a latent space into another. Doing this can transform any physics-based simulation into our database's more realistic earthquake time history. The loss function reads:

$$\begin{aligned}
&l^{h}\left(\mathbf{z},\mathbf{z}',\mathbf{x},\mathbf{y}\right) = l^h_y(\mathbf{x},\mathbf{y}) + l^h_x(\mathbf{x},\mathbf{y}) + l^h_{z'}\left(\mathbf{x},\mathbf{z}'\right) + l^h_z(\mathbf{z},\mathbf{y}) + \\
&\lambda l_{R1}\left(\mathbf{y}, G_y \circ G_z \circ F_z \circ F_y(\mathbf{y})\right) + l_{R1}\left(\mathbf{x}, G_x \circ F_z \circ F_y(\mathbf{x})\right)
\end{aligned} \tag{4.6}$$

Expression in which the different terms can be further unraveled as:

$$\begin{aligned}
l^h_y(\mathbf{x},\mathbf{y}) &= D^h_y(\mathbf{y})\mathbf{1}_{C=\text{"}d\text{"}} - D^h_y\left(G_y \circ G_z \circ F_x(\mathbf{x})\right)\mathbf{1}_{C=\text{"}d_x\text{"}} \\
l^h_x(\mathbf{x},\mathbf{y}) &= D^h_x(\mathbf{x})\mathbf{1}_{C=\text{"}d_x}\text{"} - D^h_x\left(G_x \circ F_z \circ F_y(\mathbf{y})\right)\mathbf{1}_{C=\text{"}d\text{"}} \\
l^h_{z'}\left(\mathbf{x},\mathbf{z}'\right) &= D^h_{z'}\left(\mathbf{z}'\right)\mathbf{1}_{C=\text{"}g_{z'}} - D^h_{z'}\left(G_z \circ F_x(\mathbf{x})\right)\mathbf{1}_{C=\text{"}d_x} \\
l^h_z(\mathbf{z},\mathbf{y}) &= D^h_z(\mathbf{z})\mathbf{1}_{C=\text{"}g_z}\text{"} - D^h_z\left(F_z \circ F_y(\mathbf{y})\right)\mathbf{1}_{C=\text{"}d''}
\end{aligned} \tag{4.7}$$

Equation 4.7 expresses Wasserstein Earth Mover (EM) Loss, as described in Sections 2.5, 2.5.5 and Table 2.6. Here, with l^h_y we force hybrid synthetic data $\mathbf{x}$, *i.e*, $G_y \circ G_z \circ F_x(\mathbf{x})$ to resemble to $\mathbf{y}$, forcing the encoders, and decoder to mimic a simple close to the $p(\mathbf{y})$ distribution. In the same way, we are forcing, G_x, F_z and F_z to learn the mapping between $\mathbf{x}$ and $G_x \circ F_z \circ F_y(\mathbf{y})$. $\mathbf{z}'$ is the latent space adapted that should capture the relevant features of $\mathbf{x}$ And so do $\mathbf{z}$ for $\mathbf{y}$, as illustrated in Figure 4.2.

4.1.2 Limitation of this strategy

(i) **Sequential training**
The SeismoALICE method, applied in a tripartite manner, faces limitations due to its sequential nature. Training the broadband generators and filtered data networks must precede the training of the hybrid encoder and decoder, introducing a time delay.

(ii) **Duplicate resources**
Furthermore, the approach duplicates resources, leading to an augmented number of discriminators (D_x and D^h_x, D_y and D^h_x, D_z and D^h_z). Calibration of these discriminators for generators is time-consuming and requires rather intricate training hacks, as discussed by Lee *et al.* [165]. The interplay between physics-based and seismic data introduces unnecessary parameters, thereby complicating the pursuit of optimal solution searches. Attaining a Nash equilibrium between generators and discriminators proves challenging, amplifying the complexity of parameter optimization.

(iii) **Interpretability of the latent spaces**
A critical drawback is found in the final layer of Generators when CNN is adopted as a neural network (see Gatti and Clouteau, 2020 [21]). This layer struggles to capture broad contextual information, diminishing overall model performance. The network's limited ability to project data into a Gaussian distribution preserves data correlations, necessitating exploring alternative layer structures to overcome these constraints. In the encoders, F_x or F_y, CNN extracts hidden features; even for the higher level, we will need a transformation to shift to the targeted latent space distribution (Uniform or factorized Gaussian). Such a thing is possible through the transition matrices of the Linear Layer or with the regularization technique provided by V.A.E, see Section § 2.3.2. In the case of the application of a Linear Layer on top of CNN, the last output of the CNN is flattened, and the projection proceeds, aggregating and encoding into a more compact representation, as demonstrated by Aggarwal *et al.*, 2018 [135]. It is noteworthy to talk about the advancement made in BigBigGAN about the difficulty of projecting data into a latent distribution $p(\mathbf{z})$; the work of Donahue *et al.*, 2019 [173] proposes to reuse a classifier as encoder to target a distribution $\mathcal{N}(\mu, \sigma)$. To do this, they have forced through the last layer to output the mean, μ and an estimation of the standard deviation and $\hat{\sigma}$, A last non-negative activation softplus is applied, namely softplus $= \log(1 + \exp(\hat{\sigma})$ and $\mathbf{z}$ sampled as $\mathbf{z} = \mu + \epsilon\sigma$, where $\epsilon \sim \mathcal{N}(0, I)$. Donahue has also compared it with other projection techniques to highlight issues.

(iv) **Contradictory loss**
There are two ways to preserve cycle consistency: implicitly, with an adversarial loss, or explicitly with a ℓ^2 loss. Theoretically, according to Li et al., 2017 [175], implicit consistency is equivalent to explicit consistency. However, cycle consistency loss can lead to contradictory situations and does not always preserve consistency in the context of ALICE. One might think that the auto-encoder architecture is extremely efficient. This is easy to see if the separately trained auto-encoder provides satisfactory reconstruction quality. In this case, the expected behavior when training with implicit cycling will be a deterioration of the reconstruction during training; on the other hand, it will be observed that generation under Gaussian noise is widely general. This implies that the Nash equilibrium can easily be broken if the capacities of the discriminator providing the cycling property are not sufficiently restricted. This complexity disappears with explicit cycling through standard ℓ^2 loss. Also, This strategy is more sensitive to the nature of the dataset. The tuning or arrangement of discriminators is a potential bottleneck in the process. Belghazi *et al.*, 2021 [305] in their work highlights the challenges of computing mutual information, particularly in the case of the Information Bottleneck (IB). The IB is a method that quantifies the mutual information between two variables X, Y through the mapping $X \to Z \to Y$ for the triplet $(X, Y, Z) \in (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$. *"An optimal representation of X should capture the relevant factors and compress X by reducing the irrelevant parts that do not contribute to the prediction of Y", [305]. This is done by minimizing the Lagrangian :*

$$\mathcal{L}[q(\mathbf{z}|\mathbf{x})] = H(\mathbf{y}|\mathbf{y}) + \beta I(\mathbf{x}, \mathbf{z}) \tag{4.8}$$

Formula in which, $q(\mathbf{z}|\mathbf{x})$ is the encoder outputting $\mathbf{z}$, $H(\mathbf{y}|\mathbf{z})$ is the conditional entropy, defined as :

$$H(\mathbf{y}|\mathbf{z}) = -\iint_{\mathbf{y}\times\mathbf{z}} p(\mathbf{y}, \mathbf{z}) \log \frac{p(\mathbf{y}, \mathbf{z})}{p(\mathbf{z})} dp(\mathbf{y}; \mathbf{z}) \tag{4.9}$$

$I(\mathbf{x}, \mathbf{z})$ is the mutual information between $\mathbf{x}$ and $\mathbf{z}$, whose definition reads (Tishby *et al.*, 2015 [306]):

$$I(\mathbf{x}, \mathbf{z}) = \iint_{\mathbf{x} \times \mathbf{z}} \log \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})p(\mathbf{z})} dp(\mathbf{x}, \mathbf{z}) \tag{4.10}$$

Where $p(\mathbf{x}, \mathbf{z})$ is the joint probability distribution, $p(\mathbf{x})$ and $p(\mathbf{z})$ are the marginal distribution, the definition of mutual information corresponds to the Kullback-Leibler divergence, D_{KL}. This expression is *a priori* intractable. A parametrized neural network could be used to estimate this expression, $\mathcal{F} = \{T_\theta\}_{\theta \in \Theta}$:

$$I(\hat{\mathbf{x}, \mathbf{z}}) = \sup_{\theta \in \Theta} \mathbb{E}_{p(\mathbf{x}, \mathbf{z})}[T_\theta] - \log \mathbb{E}_{p(\mathbf{x})q(\mathbf{z})}[e^{T_\theta}] \tag{4.11}$$

The IB corresponds when we search for the balance between two conflicting objectives in machine learning. Thereby, the complexity becomes problematic. The use of ℓ^2 can at least ensure reconstruction. A more efficient architecture for the generators could be envisaged to improve this quality, as shown in the next sections.

One primary objective of this thesis is to solve the problem $\mathcal{P}_y$ to attain high-fidelity reconstruction of broadband signals and generate the whole database from Gaussian noise, $\mathbf{z}_y \sim \mathcal{N}(0, \mathbf{I})$, considering the 0-30 Hz frequency interval. Each 3-component broadband signal exhibits a distinct physical behavior governed by a coherent and sequential progression, commencing with the P-wave, succeeded by the S-wave. Consequently, the signal displays long-range dependencies, rendering it a complex challenge to analyze. This distinct seismic signal pattern consistently manifests within our dataset (STEAD). Therefore, the foremost step in our research involves identifying the most suitable architecture to handle this task effectively. To initiate our approach, we conduct a meticulous investigation to determine the optimal encoder and decoder, ensuring the highest precision in data reconstruction.

Subsequently, we introduce an adversarial loss to augment the quality of the reconstruction. By employing adversarial training techniques, a discriminator is incorporated to enhance the fidelity of the generated data. Our overarching objective involves projecting the physics-based data into a Gaussian distribution and retrieving the corresponding encoded signal. Consequently, the trained Generator can generate the complete database from randomly generated noise, leveraging the encoded representation as a guiding blueprint. Through these meticulous steps, we endeavor to achieve superlative reconstruction results and establish a more authentic representation of the original broadband signals.

SeismoALICE recap

SeismoALICE is a deep-CNN-based GAN to render synthetic broad-band ground motion time-histories (0-30 Hz) from low-frequency simulated accelerograms (0-1 Hz). The task is achieved by encoding/decoding both numerical simulations $\mathbf{x}$ (frequency band: 0-1 Hz) and corresponding recorded seismograms $\mathbf{y}$ (frequency band: 0-30 Hz) into Gaussian latent space $\mathbf{z}$ and $\mathbf{z}'$ respectively. The goal is achieved by adopting the Adversarial Learning Inference with Cross-Entropy (ALICE), proposed by Dumoulin *et al.*, [304] 2017. The goal of this chapter is to improve the original version of SeismoALICE to achieve the following milestones:

(i) Training sets of encoder/Decoder with a good reconstruction quality for both $\mathbf{x}$ (low-frequency input) and $\mathbf{y}$

(ii) The encoder can project any 3-component seismogram data in 1-dimensional Gaussian vectors

(iii) Generate 3-component seismograms from the 1D Gaussian vector

4.2 Improving SeismoALICE

Generators

(i) **auto-encoder structure (AE)**
We directly start by seeking to optimize the reconstruction quality through an AE architecture. The task we want to perform in times series needs an architecture capable of capturing long-range dependencies. The original SeismoALICE adopted encoders and decoders, conceived as stacks of 1D convolutional layers. In the previously presented architecture, the flow of the information passes through a stacked convolution block without researching a way to prevent the model from forgetting relevant information. To circumvent this problem, a residual block, with skip-connection, is adopted in the architecture ConvResBlock and ConvBlockTranspose. See the Figure 4.4 for reference.
The ConvResBlock comprises two branch orders: one for normal features extraction and the other for skip-connection. Instance Norm, ReLU, Dropout, Convolution with stride 2, and Convolution element-wise are used.
For the ConvBlockTranspose, instead of using convolutions transposed, which could have a chessboard effect, we use the convolution element-wise, but the upscaling is the Interpolation with factor 2. This architecture is inspired by SAGAN (Self-Attention Generative Adversarial Networks), Zhang *et al.*, 2019 [220]. Moreover, we employ an attention mechanism. Our investigation has led us to Conformers (view § 2.4.6), which we use in the encoder and decoder architecture. Due to the computational cost, this employment is limited to two for the encoder and two for the decoder. We normalized the layers' weights by using spectral normalization, Miyato *et al.*, 2018 [307], as a regularization technique at all levels of the architecture, which improves Generator stability and reinforces the Lipschitz continuity for the training for the Discriminators (more exploration is the sections bellow), according to Brock *et al.*, 2019 [185]. Zhang *et al.*, 2019 [220] advocate further that, *"Spectral normalization in the generator can prevent the escalation of parameter*

magnitudes and avoid unusual gradients.”. The formulation of the Special Normalization according to Miyato *et al.*, 2018 has for expression for a matrix A with ℓ^2 norm on A:

$$\sigma(A) := \max_{\mathbf{h}:\mathbf{h}\neq\mathbf{0}} \frac{\|A\mathbf{h}\|_2}{\|\mathbf{h}\|_2} = \max_{\|\mathbf{h}\|_2\leq 1} \|A\mathbf{h}\|_2 \tag{4.12}$$

A spectral normalization of the weight matrix $\mathbf{W}$ as therefore per expression :

$$W_{\text{SN}}(\mathbf{W}) := \frac{\mathbf{W}}{\mathbf{W}} \tag{4.13}$$

In addition, to normalize the layers, we perform the normalization of the output, which is replaced by the Instance Norm, for the advantages detailed in Section § 3.2.3.

(ii) **Choice of bottleneck layer**

The specific aspect of the AE is the bottleneck; the latent values should condense all the relevant features, but at the same time, they should be large enough to provide good reconstruction for the times series. We have tested different 1D-vector sizes of 64, 128, and 256 values. The encoder’s final layer, called “bottleneck”, is crucial because it will project the feature maps into a 1D flat vector of Gaussian variables. In the original SeismoALICE formulation, the bottleneck is a 1D CNN layer, which outputs a sequence of feature maps consisting of multi-variate time histories instead of outputting a 1D flat vector of Gaussian variables. This preserves the initial signals’ temporal causality but does not represent the best compression solution. To further reduce the order of the latent manifold, several other options were tested, including Fourier Neural Operator (FNO)[308, 309, 264] and dense feed-forward architectures, which represent the standard solution for images. The FNO was proposed to replace the kernel operation of convolution to operate in Fourier space [308, 309]. If $\mathcal{F}$ is the Fourier transform of a function $f : D \to \mathbb{R}^{d_v}$ and $\mathcal{F}^{-1}$ it inverse, then :

$$(\mathcal{F}f)_j(k) = \int_D f_j(x) e^{-2i\pi\langle x,k\rangle} \mathrm{d}x, \qquad (\mathcal{F}^-1f)j(x) = \int_D f_j e^{2i\pi\langle x,k\rangle} \mathrm{d}k \tag{4.14}$$

for $j = 1, \cdots, d_v$, $i = \sqrt{-1}$, the imaginary unit. A kernel is defined by:

$$\left(\mathcal{K}(a;\phi)v_t\right)(x) := \int_D \kappa(x, y, a, a(x), a(y); \phi) v_t(y) dy, \quad \forall x \in D \tag{4.15}$$

The term κ_ϕ is a neural network parameterized by ϕ, $\kappa_\phi : \mathbb{R}^{2(d+_a)\to\mathbb{R}^{d_v\times d_v}}$, $\phi \in \Theta_\mathcal{K}$. It has been demonstrated that,

$$\kappa(x, y, a, a(x), a(y); \phi) = \kappa(x - y; \phi) \tag{4.16}$$

The final expression of the Fourier integral operator $\mathcal{K}$ has for expression:

$$(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}(\mathcal{F}(\kappa_\phi) \cdot \mathcal{F}(\kappa_\phi))(x) \quad \forall x \in D \tag{4.17}$$

a is function coming from the Banach space of dimension $\mathbb{R}^{d_a}$. Our investigation as leverage that the FNO applying at the last layer of the encoder could help project to a desired latent distribution space, $p(\mathbf{z})$, illustration of FNO could be found in Figure 4.3. However. this strategy introduces a trace of correlation and could not provide a compact representation, and then $\mathbf{z} \sim p(\mathbf{z})$ could not target back the data distribution space,

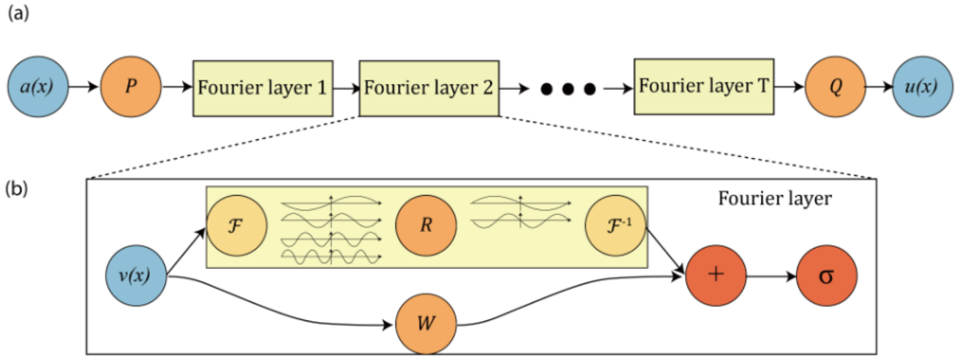


Figure 4.3: ©[308], the full architecture of Fourier Neural Operator. *"(a) The full architecture of neural operator: start from input* a*. 1. Lift to a higher dimension channel space by a neural network* P*. 2. Apply four layers of integral operators and activation functions. 3. Project back to the target dimension by a neural network* Q*. Output* u*. (b) Fourier layers: Start from input* v*. On top: apply the Fourier transform* $\mathcal{F}$*; a linear transform* R *on the lower Fourier modes and filters out the higher modes; then apply the inverse Fourier transform* $\mathcal{F}^{-1}$*. On the bottom: apply a local linear transform* W*."*

$p_{\text{data}}(\mathbf{x})$. Therefore, dense layers are used at the end of the encoder and the decoder's beginning. More explanation about that choice is criticized in Section 4.1.2. The dense feed-forward architecture is the most suitable for encoding original time-series data into a Gaussian vector. This choice was also driven by its efficacy in preserving essential features during the projection process.

(iii) **Architecture variant proposed**
After many tests, we found that architecture with Attention Technique, especially Transformer or Conformer, is the most appropriate to capture long-range dependency for the times series, particularly for the broadband signals. A mix of Conformer + UNet does not significantly improve the quality of the reconstruction, which already has a "good" quality according to the G.O.F. listed in Table 2.8. The architecture of the generators with Conformers is presented in Figure 4.4. We use an encoder and a decoder that incorporate in this architecture Conformer; the choice of bottleneck layer is the Linear Layer, among other type of architecture we have experienced (encoder with convolution bottleneck layer, with FNO bottleneck layer). This makes a deeper analysis of the data and extracts appropriate features from deeper layers (see Figure 4.4). The Conformer is efficient in the penultimate layer. Other placements have shown some weird behavior and the representation of the signal. See in the Table 4.1 and Table 4.2 a description about the architecture.

(iv) **Comparison with previous architecture**
Compared to Gatti *et al.*,2020 [21], our architecture reduces the number of hidden variables to translate from broadband to physic-based and vice versa. Then, a better understanding of the interpretability of the latent values is provided. Our architecture can perform generation through the adequate choice of projection technique to latent space. The introduction of diverse regularization techniques drastically reduces the number of loss functions to guarantee the previous assumption while preserving stability and the Lipschitz continuity. In the previous technique, the Lipschitz was forced through weigh clipping, which penalized the efficiency of the workflow.

Layer	**Opeation**	**Input Size**	**Fonction Activation**	**Stride**	**Output**
ConvBlock I	Downsampling	[3, 4096]	LeakyReLU	2	[16, 2048]
ConvBlock II	Downsampling	[16, 2048]	LeakyReLU	2	[32, 1024]
ConvBlock III	Downsampling	[32, 1024]	LeakyReLU	2	[64, 512]
ConvBlock IV	Downsampling	[64, 512]	LeakyReLU	2	[128,256]
Conformer	Attention	[128,256]	’–	’–	[128,256]
ConvBlock V	Downsampling	[128,256]	LeakyReLU	2	[256,128]
ConvBlock VI	Downsampling	[256,128]	LeakyReLU	2	[512,64]
Conformer	Attention	[512,64]	’–	’–	[512,64]
Reshape	’–	[512,64]	’–	’–	[32768]
Linear Layer	Projection	[32768]	’–	’–	[256]

Table 4.1: Architecture of the Encoder

Layer	**Opeation**	**Input Size**	**Fonction Activation**	**Stride**	**Output**
Linear	Projection	[256]	’–	’–	[32768]
Reshape	’–	[32768]	’–	’–	[512,64]
ConvTransBlock I	Upsampling	[512,64]	LeakyReLU	2	[256, 128]
ConvTransBlock II	Upsampling	[128, 128]	LeakyReLU	2	[64, 256]
Conformer	Attention	[64,256]	LeakyReLU	2	[64, 256]
ConvTransBlock III	Downsampling	[64, 256]	LeakyReLU	2	[32, 1024]
ConvTransBlock IV	Attention	[32, 1024]	’–	’–	[16, 2048]
ConvTransBlock V	Downsampling	[16, 2048]	LeakyReLU	2	[16, 2048]
Conformer	Attention	[16,2048]	LeakyReLU	2	[16,2048]
ConvTransBlock VI	Downsampling	[16,2048]	Tanh	’–	[3,4096]

Table 4.2: Architecture of the Decoder for SeismoALICE

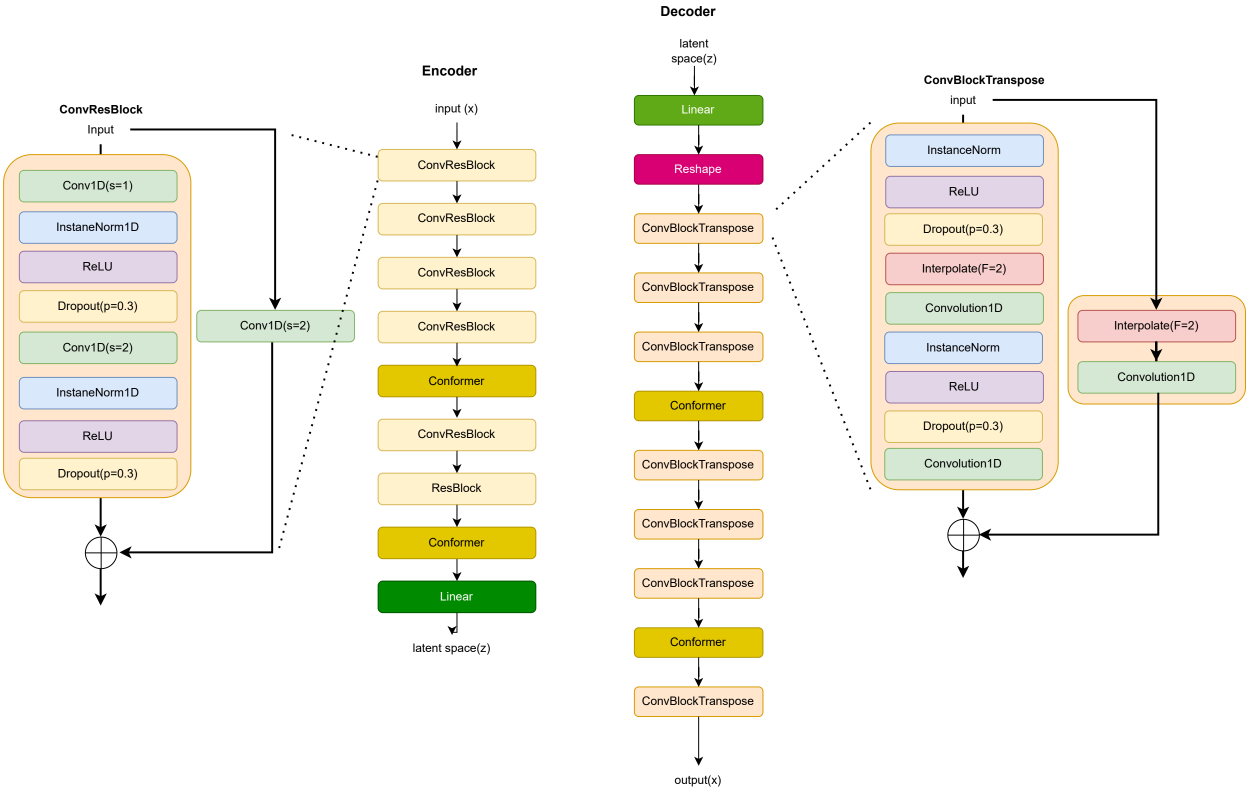


Figure 4.4: Generators with conformer architectures. The ConvResblock is composed of a residual Block. The ConvTransposeBlock does not use ConvTranspose1d, but the upsampling is done using interpolate with factor 2. A description of ConvResBlock and ConvTransposeBlock architecture can be found in section §4.2.

Discriminators

The formulation of Discriminator posed another significant challenge, particularly due to its pivotal role in the comprehensive training process. In the complete training, the generators (encoder and decoder) undergo updates based on the evaluation provided by the discriminator. So the design of the discriminator should be coherent and efficient, capturing and not destroying the information, even though the main task in the complete training is to fool the discriminator [80, 290, 231]. To this end, the original SeismoALICE discriminators were condensed into a unified architecture with three branches. This occurred for both joint discriminators, for $(\mathbf{x}, \mathbf{z})$ and $(\mathbf{y}, \mathbf{z}')$. For the sake of clarity, we described the three branches only for the pair $(\mathbf{y}, \mathbf{z}')$, but the same architecture was adopted for $(\mathbf{x}, \mathbf{z})$. The first one, named D_{sy}, has the task to extract the relevant signature of the signal $\mathbf{y}$ or $G_y(\mathbf{z}_y)$. We adopted the same architecture of the encoder to design D_{sy}. This proposed design was advised by Brock *et al.*, 2019 [185], *"As far as the network configuration is concerned, the discriminator is an exact reflection of the generator."*. The second branch, D_{szb} has the task to discriminate between $\mathbf{z}_y$ and $F_y(\mathbf{z}_y)$. D_{sy} and D_{szb} can be considered as logits values or intermediate feature maps that are concatenated and fed into the discriminating branch $D_{yz}(D_{sy}(\mathbf{y}), D_{szb}(\mathbf{z}_y))$. D_{yz} is naturally featured by a final sigmoid activation function, $\sigma(\bullet)$, that should output a value between 0 and 1 (a probability of the sample to be true or false). By experience, the BCEWithlogit[1](Binary Cross Entropy With Logits) is numerically more stable than the classic BCE[2](Binary Cross Entropy). This same output is suitable for Hinge Loss, proposed in GAN by [185], which is treated in Section 4.3, this is our choice. Figure 4.5 depicts the scheme of the unified discriminator architecture. Therefore, the retained ALI loss has pour expression for the broadband by example:

$$\mathcal{L}_{\text{ALI}} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\log(\sigma(D_{yz}(\mathbf{y}, F_y(\mathbf{y}))))\right] + \mathbb{E}_{\mathbf{z}'\sim p(\mathbf{z}')}\left[\log(1-\sigma(D_{yz}(G_y(\mathbf{z}'), \mathbf{z}')))\right] \tag{4.18}$$

Additional Information of the architecture can be seen in Table 4.3 and Table 4.4. The architecture of D_{s_y} because is the same as Encoder F_y (explained above). In addition, to fool the discriminator easily and foster generator training, we carefully calibrated the noise. Aggarwal *et al.*,2018 [135] has demonstrated that *"The addition of noise to the input has connections with penalty-based regularization"*. Clamping a white noise $\mathbf{n}$ to $\mathbf{y}$, *i.e.*, $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{n}$, with values in the range -0.5 and 0.5. It is noteworthy, to be precise, that $\mathbf{y}$ is normalized by default within -1 and +1. The choice of the value of the clamp is a trade-off, and it might change depending on the dataset. The clamp reduces the amplitude of the white noise, polluting the original data $\mathbf{y}$, particularly for the high frequencies. A quick overview of the frequency domain of synthetic data will showcase that high-frequency data is different from broad-band data. This pathology was already present in Gatti *et al.*, 2020 [21]. Therefore, we must make it difficult for the discriminator to penalize high frequency regarding numerical limitations. Therefore, the introduction of that noise stabilizes the training. Moreover, after a thorough investigation, the most appropriate discriminator should incorporate the residual block of the neural network and spectral normalization technique (see Figure 4.5) for better stabilization. Stable training, defined as opposed to the pathology that GAN may present, corresponds to the fact

---

[1] The mathematical expression of cross entropy between two variable x and y is defined by: $\ell(\mathbf{x}, \mathbf{y}) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_n\left[y_n \cdot \log\sigma(x_n) + (1-y_n)\cdot\log(1-\sigma(x_n))\right]$. In this formulation we have $\mathbf{x} = x_1, x_2, \cdots, x_n$ is the set and $\mathbf{y} = y_1, y_2, \cdots, y_n$ are the labels, usually $\mathbf{y} = \mathbf{1}$ or $\mathbf{0}$. σ correspond to the sigmoid. In the case of machine learning, this corresponds to the fact that this activation is not present in the last discriminator layer. The numerical stability of this expression has been advocated. The w_n is the weight of importance of the loss

[2] The expression of BCE is defined by:
$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_n\left[y_n \cdot \log x_n + (1-y_n)\cdot\log(1-x_n)\right]$

Layer	**Opeation**	**Input Size**	**Fonction Activation**	**Stride**	**Output**
ResBlockDense I	Downsampling	[512]	LeakyReLU	2	[512]
ResBlockDense II	Downsampling	[512]	LeakyReLU	2	[512]
ResBlockDense III	Downsampling	[512]	LeakyReLU	2	[512]
Linear Layer	Projection	[512]	'–	'–	[1]

Table 4.3: Architecture of the Discriminator $D_{s_{yz}}$

Layer	**Opeation**	**Input Size**	**Fonction Activation**	**Stride**	**Output**
ResBlockDense I	Downsampling	[256]	LeakyReLU	2	[256]
ResBlockDense II	Downsampling	[256]	LeakyReLU	2	[256]
ResBlockDense III	Downsampling	[256]	LeakyReLU	2	[256]
Linear Layer	Projection	[256]	'–	'–	[256]

Table 4.4: Architecture of the Discriminator $D_{s_{zy}}$

that training over time moves towards Nash equilibrium without deviating significantly from its objective and without producing undesirable outcomes, Becker *et al.*, 2022 [310]. GAN issues are detailed in section Section 2.3.3. This is applied to the convolutions and for fully-connected layers. The last layer converts the features into a scalar, scoring, not probability. The BCEWithlogit is used for numerical stability instead of classical BCE.

Optimization and Regularization

(i) **Optimization loss**

The optimization is done through the loss of Focal Frequency and the hyper-spherical loss. After many tests, we kept the Conformer architecture with a bottleneck dimension 256. The result is not significantly different from the dimension 128, but the size 256 is tailored for the hybrid mapping, which we will train later. View Figure 4.6. For the sake of performance, we have trained our loss function with different adversarial objectives, with HingeLoss (View Equation 4.19) or Least Square Mean Loss (View Equation 4.20) or Earth Motion Loss (Wasserstein). We found that the result is not significantly different between BCEWithlogit. The variant of adversarial loss with HingeLoss has an equation:

$$\begin{aligned} \mathcal{L}_{\mathcal{D}} &= \mathbb{E}_{y\sim P_{data}}\left[\max(0, 1 - D_{yz}(\mathbf{y}, F_y(\mathbf{y})))\right] + \mathbb{E}_{z\sim P_z}\left[\max(0, 1 + D_{yz}(G_y(\mathbf{z})), \mathbf{z}_y)\right] \\ \mathcal{L}_{\mathcal{G}} &= \mathbb{E}_{y\sim P_{data}}\left[D_{yz}(\mathbf{y}, F_y(\mathbf{y}))\right] + \mathbb{E}_{z\sim P_z}\left[-D_{yz}(G_y(\mathbf{z}), \mathbf{z}_y)\right] \end{aligned} \tag{4.19}$$

For Least Square Mean Loss :

$$\begin{aligned} \mathcal{L}_{\mathcal{D}} &= \mathbb{E}_{y\sim P_{data}}\left[(D_{yz}(\mathbf{y}, F_y(\mathbf{y})) - 1)^2\right] + \mathbb{E}_{\mathbf{z}_y\sim P_{\mathbf{z}_y}}\left[(D_{yz}(G_y(\mathbf{z}_y), \mathbf{z}_y))\right] \\ \mathcal{L}_{\mathcal{G}} &= \mathbb{E}_{y\sim P_{data}}\left[(D_{yz}(\mathbf{y}, F_y(\mathbf{y})) - 1)^2\right] + \mathbb{E}_{\mathbf{z}_y\sim P_{\mathbf{z}_y}}\left[(D_{yz}(G_y(\mathbf{z}_y), \mathbf{z}_y) - 1)^2\right] \end{aligned} \tag{4.20}$$

The Earth Motion (EM) Loss seems not properly adapted (WGAN and WGAN-GP) for our case.

The cycle consistency is made explicitly with FFL and HSL :

$$\mathcal{L}_{\text{Cycle}}(F_y, G_y, \mathbf{y}) = \text{FFL}(\mathbf{x}, G_y(F_y(\mathbf{y}))) + \text{HSL}(\mathbf{x}, G_y(F_y(\mathbf{y}))) \tag{4.21}$$

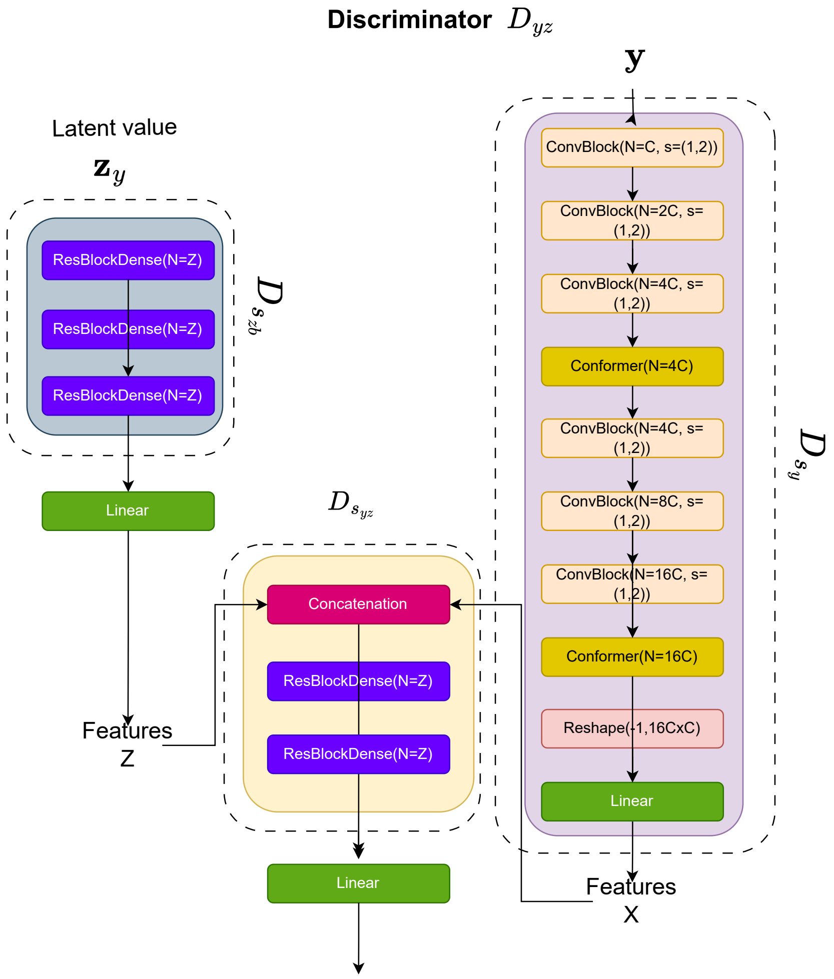


Figure 4.5: Architecture of the joint discriminator of D_{yz}. The Design includes a Residual Block. $C = 64$, N represent the number of Channels, s represent the strides. The number of parameters of the discriminator is 10 million parameters. This architecture comprises two branches, D_{s_x} and $D_{s_{zy}}$.

(ii) **Regularization technique**

We tested different weight regularization techniques designed to stabilize the training (Spectral Norm[307], Weight Orthogonality [311], Weight Norm [312]). *"The weight normalization methods have been employed for satisfying Lipschitz continuity while the weight penalty methods have been introduced for orthogonality of weight matrices or general stability of the GAN training. All the weight normalization methods have been reported that they commonly enhance sample generation performance of GANs"* [165]. The Spectral normalization is formulated in Equation 4.13.

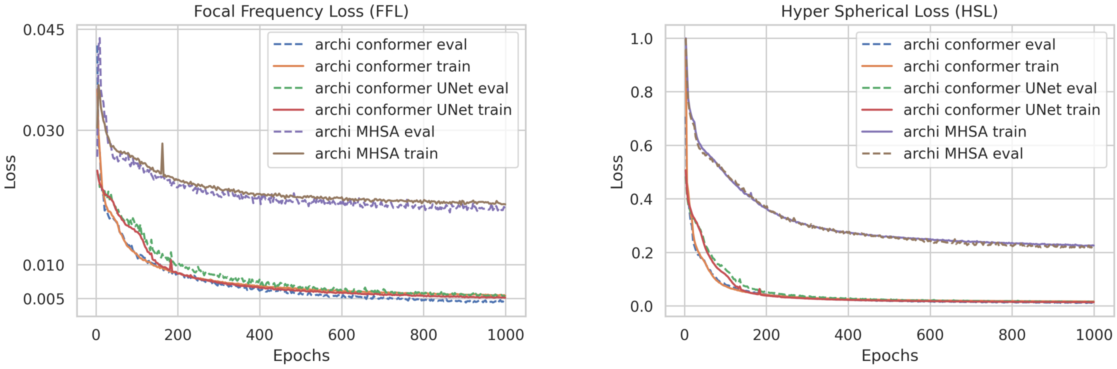
Brock formulates the Orthogonal regularization as:

$$R_\beta(\mathbf{W}) := \beta \|\mathbf{W}^\top \mathbf{W} - \mathbf{I}\|_F^2 \tag{4.22}$$

Formula in which $\|\cdot\|_F$ is the Forbinius norm, $\mathbf{I}$ the identity matrix. β is a hyper parameter, commonly choose with 10^{-4}.

The Weight Norm technique is a reparametrization method that decouples the magnitude of the weight from its directions. Assuming a neural network operation is expressed by :

$$y = \phi(\mathbf{w} \cdot \mathbf{x} + b)$$



(a) FFL for the unitary test of broadband with different encoder/decoder architectures.

(b) HSL values range between 1.0 and 0. A lower HSL indicates better reconstruction.

Figure 4.6: General investigation of architectures adapted for time series in Broadband signals. The comparison of different losses highlights the advantage of using the Conformer architecture for adversarial training. The HSL can be considered as a metric.

Model Architecture	**# of param.(M)**	**EG**	**PG**
Mutli Head Self Attention	32	2.34 ± 1.33	6.94 ± 0.55
Conformer	114	7.06 ± 0.60	8.60 ± 0.35
Conformer+Unet	118	6.98 ± 0.77	8.72 ± 0.35

Table 4.5: Table of Architecture of Generators for the research of the best adatped model for reconstruction of data. The Data set contains 128,000 3D signal. 10% is used as the validation and 10% serves as test.

ϕ is an activation functions , $\mathbf{w}$ is the k-dimensional matrix, $\mathbf{x}$ a k-dimensional vector of input. The weight regularization consists of the research of a parameter vector $\mathbf{v}$ and a scalar g, such a way that:

$$\mathbf{w} = \frac{g}{\|\mathbf{v}\|}\mathbf{v} \tag{4.23}$$

The norm of $\mathbf{w}$ is independent of $\mathbf{v}$ and equals g. Other techniques were listed by Lee *et al.*,[165]. As highlighted by Kumar *et al.*, 2019 [231], even though the normalization technique should improve the training, some weight normalization may work better depending on the data set. In our case, the solution that works is the normalization of the weight of the layers with Spectral Normalization.

(iii) **Mode collapse and Mode Dropping solutions**
We train either Early Stopping or CyclicLR to avoid mode collapse. Another common form of regularisation is early stopping, which stops the gradient descent after a few iterations when the error on a retained test set increases. Early stopping is a regularizer because it effectively restricts the parameter space [135]. The CycleLR (cyclical learning rate), Smith *et al.*, 2017 [313], is a learning scheduler that makes the learning rate oscillate between two values. This strategy was used for fast convergence and increased performance as a scheduler over the optimization parameters of the discriminator.

(iv) **Training dynamics**
The ALICE training is performed using TTUR (Two Time-Scale Update Rule), proposed by Heusel *et al.*, 2018 [171]. The optimization is done using Adam optimizer. The learning

rate of the encoder/decoder is both $4 \cdot 10^{-4}$, and the learning rate of the discriminators is 10^{-4}. The data set contains 128,000 3D signals. Finally, the G.O.F serves as a metric.

The training has been made on 4 GPUs A100 with CUDA 11.7.

4.3 Results and discussion

(i) **Goodness of Fit**
After training on our improved SeismoALICE model, we evaluate the quality of the reconstruction through GOF metrics (see Section § 2.6), depicted in Figure 4.7, in terms of envelope score (EG) and phase score (PG). Figure 4.12 presents the solution for the training of broadband data set using the auto-encoder (F_y, G_y) with the architecture implementing the Conformer (See Figure 4.4(a)). As shown in Table 4.5, we have proof that architecture is adapted for cycle consistency. The means of values for EG is 7.06 and PG 8.60. In this light, given that our architecture could support the bottleneck, we superpose the Adversarial Learning Inference loss to manage the latent distribution. The GOF is also preserved in Figure 4.4(b). Therefore, the Goodness of Fit is a crucial criterion for evaluating the performance of a model.

(ii) **Cycling with adversarial loss**
It is common, in the training of ALICE, to add an adversarial loss on $\mathbf{z}'$ and $\mathbf{z}$ to force the marginal distribution of $G_y(\mathbf{y})$ and $G_x(\mathbf{x})$ to follow Gaussian distribution. This adversarial loss reads (for $\mathbf{y}$):

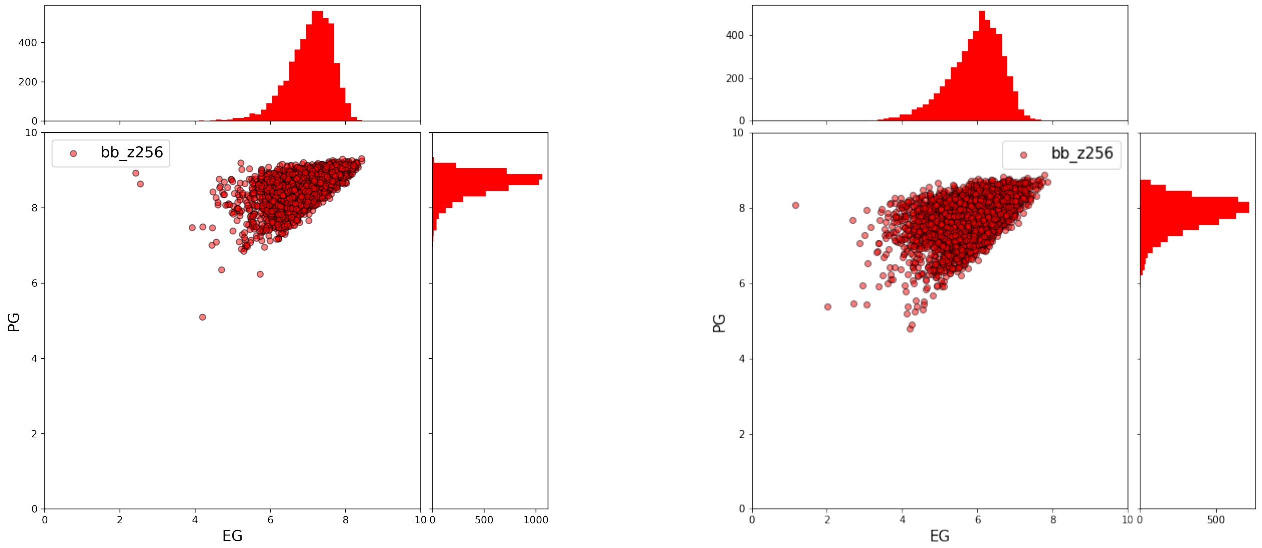
$$l_{D_{zz}} = \mathbb{E}_{z\sim p(z)}\left[\ln D_{zz}(\mathbf{z}_y, \mathbf{z}_y)\right] + \mathbb{E}_{z\sim p(z)}\left[\ln(1 - D_{zz}(\mathbf{z}_y, F_y(G_y(\mathbf{z}_y))))\right] \tag{4.24}$$

Our experiment of using the contradictory loss on the marginal distribution failed because this additional loss does not contribute to the overall convergence to the Nash equilibrium. On the contrary, the latter loss can often cause a problem. From experience, the network learns to cycle over the $\mathbf{z}_y$ without forcing the $\mathbf{z}_y$ mapper to look like an element in the database. We have, therefore, removed it. The contradictory loss on the joint distribution is sufficient. In addition, cross-entropy with contradictory loss on $\mathbf{y}$ may not work correctly depending on the dataset, as shown in Figure 4.8.

(iii) **Unconditional generation from Gaussian noise**
The FFL and Hyper Spherical Loss guarantee the reconstruction. This is evident in Figure 4.9. We can render various realistic seismograms from Gaussian noise, comparable with the training database, STEAD. The architecture captures the right casual order between P-Wave and S-Wave arrival times and the overall envelope with exponential decay toward the "*coda*" of the signal.

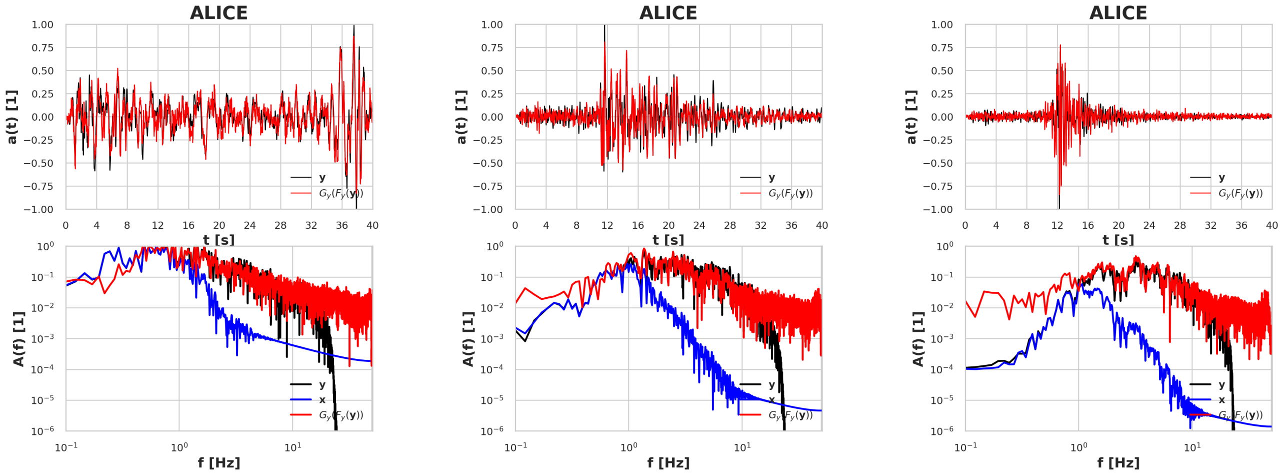
(iv) **Comparative Analysis with Existing Methods** In light of Figure 4.7, the training loss curve for ALICE using the BCEWithLogit decreases consistently within the first 1200 training epochs, thanks to the improved generator architectures. The TTUR dynamics make the training curve smooth. The theoretical optimal solution of the loss of the discriminator is $\mathcal{L}_{\mathcal{D}} = 2\log(\frac{1}{2}) \approx 1.386$ and this theoretical value is achieved using the adversarial loss of the generator, $\mathcal{L}_{\mathcal{G}}$.

Additionally, we have tested the training with HingeLoss; see the definition of this loss for GAN in Equation 4.19. As an adversarial loss, the performance that we achieve is similar.



(a) G.O.F for AE. In the case of this training, we only use the encoder, F_y, and the decoder, G_y. We have used FFL and HSL to force reconstruction, which corresponds to Cycle consistency loss. The optimisation is done according to Equation 4.21.

(b) G.O.F for ALICE. The loss includes the adversarial loss of ALI, $\mathcal{L}_{\text{ALI}}$ (Equation 4.18) and the explicit cycle consistency, $\mathcal{L}_{cycle}$ (Equation 4.21).



(c)

(d)

(e)

Figure 4.7: We present the broadband reconstruction, $G_y(F_y(\mathbf{y}))$ for AE in (a) and (b), and for ALICE in (c), (d), (e), representing different examples of reconstruction from adversarial training. $\mathcal{L}_{\text{ALI}} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\log(\sigma(D_{yz}(\mathbf{y}, F_y(\mathbf{y}))))\right] + \mathbb{E}_{\mathbf{z}'\sim p(\mathbf{z}')}\left[\log(1-\sigma(D_{yz}(G_y(\mathbf{z}'), \mathbf{z}')))\right]$ and $\mathcal{L}_{\text{Cycle}} = \text{FFL}(\mathbf{y}, G_y(F_y(\mathbf{y}))) + \text{HSL}(\mathbf{y}, G_y(F_y(\mathbf{y})))$. In the frequency domain, we observe low-frequency values that are not exactly close to the original values. Additionally, for low frequencies, we do not observe the exponential decrease for the high-frequency values greater than 20 Hz. This is a problematic observation for synthetic data that still needs to be resolved.

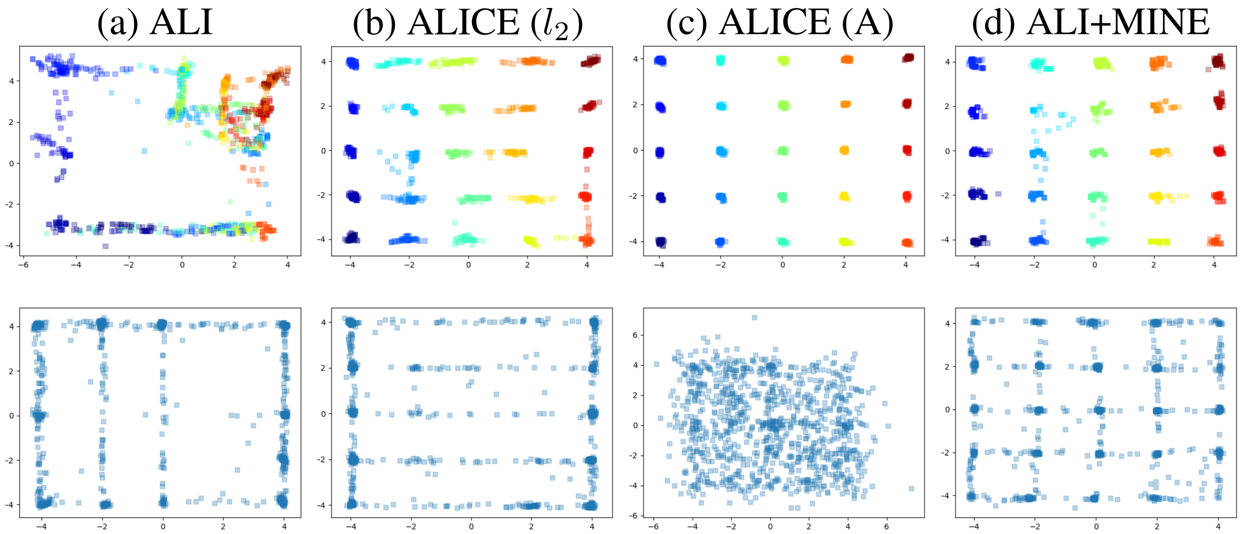


Figure 4.8: Limitation of ALICE with adversarial loss. The results show that the adversarial loss could not perform well with two different data sets, depending on the data set. The Adversarial Loss for cycle consistency could diverge the reconstruction of the signal. The ℓ^2 loss performs well for reconstruction and mapping. Source [305]

The theoretical solution for the loss of the discriminator is $\mathcal{L}_\mathcal{D} \approx 2$ and the adversarial loss for the generator is $\mathcal{L}_\mathcal{G} \approx 0$. See Figure A.1 in Annexes for more detail.

(v) **Bottleneck**

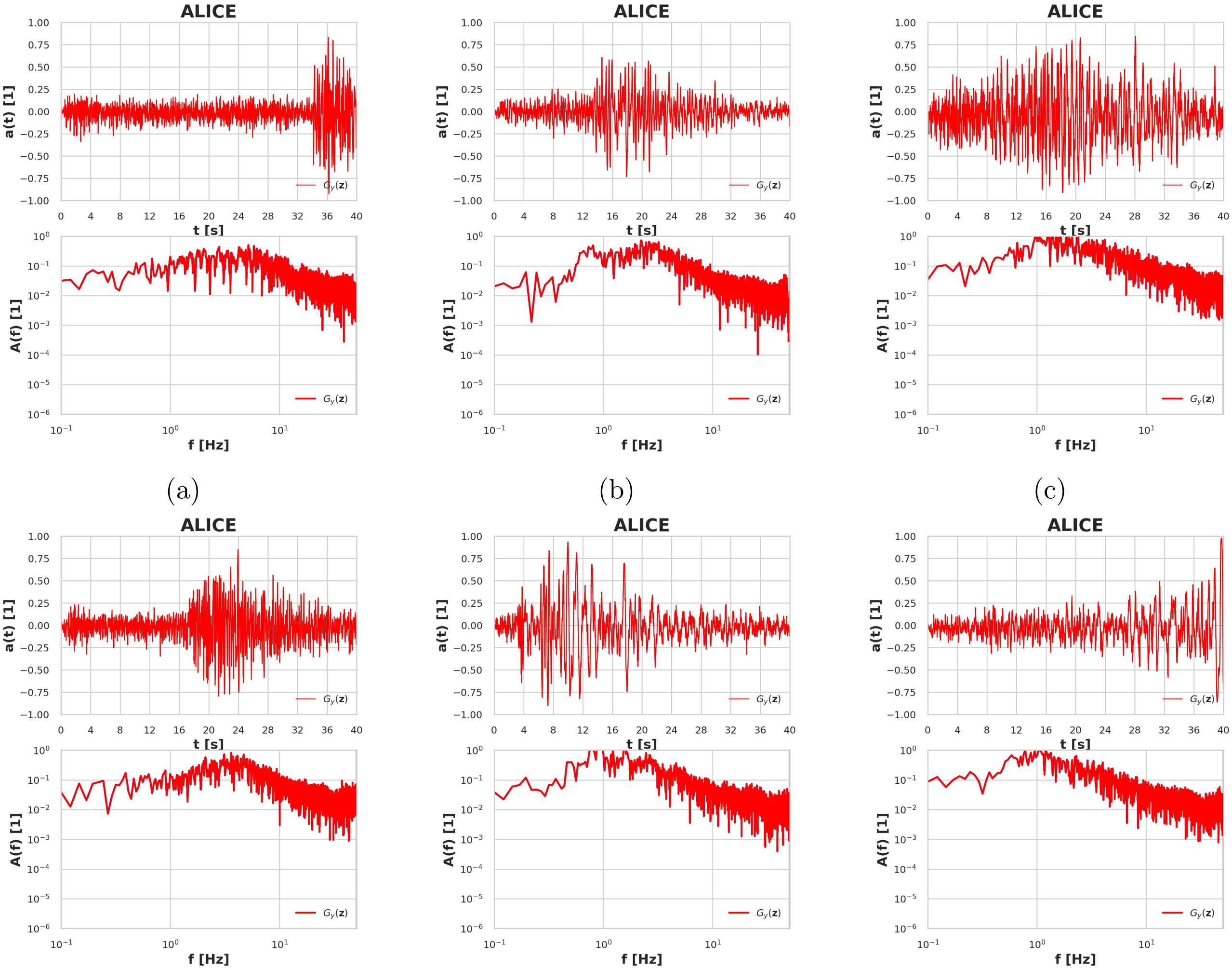
The capability of the broadband architecture lies in its proficiency to generate a broadband signal $G_y(F_y(\mathbf{x}))$ derived from the encoding of a filtered signal $\mathbf{x}$. This quality is improved by introducing the Attention Technique described in subsection 2.4.3 [119]. However, it should be noted that this does not assert that the network can precisely reconstruct the original broadband signal from which the filtered signal originates. However, according to Belghazi *et al.*, 2021 [305], the reconstruction error $\mathcal{R}$ is bounded according to the following expression: If $q(\mathbf{x})$ denotes the marginal data distribution, the reconstruction error is formulated by :

$$\mathcal{R} = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}\left[-\log(p(\mathbf{x}|\mathbf{z})\right] \tag{4.25}$$

The demonstration proposed in [305] has given as results :

$$\mathcal{R} \leq D_{KL}(q(\boldsymbol{y},\boldsymbol{z})\|p(\boldsymbol{y},\boldsymbol{z})) - I_q(\boldsymbol{y},\boldsymbol{z}) + H_q(\boldsymbol{z}) \tag{4.26}$$

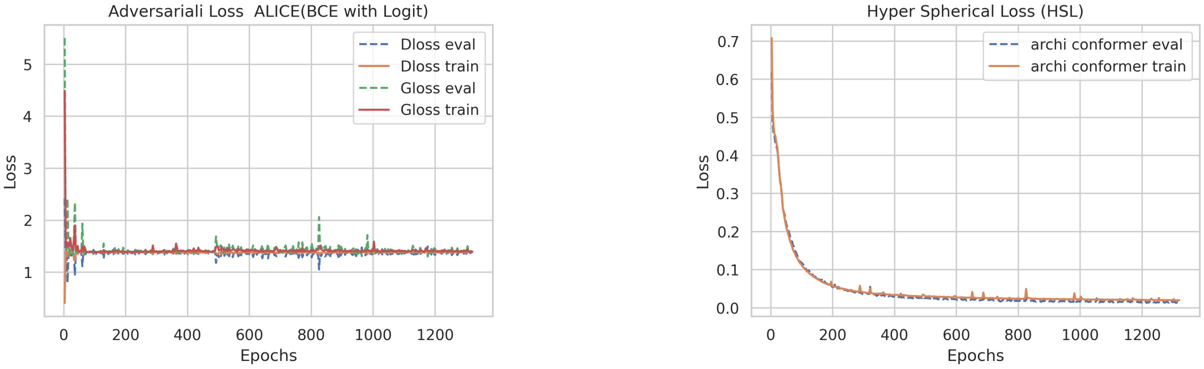
In the Equation 4.26, the terms D_{KL} is the Kullback-Leibler divergence, I_q is the mutual information, H_q is the Shannon's differential entropy. The minimization of the divergence D_{KL} through the adversarial loss is possible by trying to reduce the distances between pairs $(\mathbf{y}, \hat{\mathbf{z}})$ and $(\hat{\mathbf{y}}, \mathbf{z})$. Mutual information, I_q, quantifies the amount of information shared between variables (dependence between the two variables $\mathbf{y}$ and $\mathbf{z}$), and entropy, H_q, characterizes the uncertainty in a random variable distributed according to probability density q. Despite being computationally intractable, the entropy is an unknown constant value.



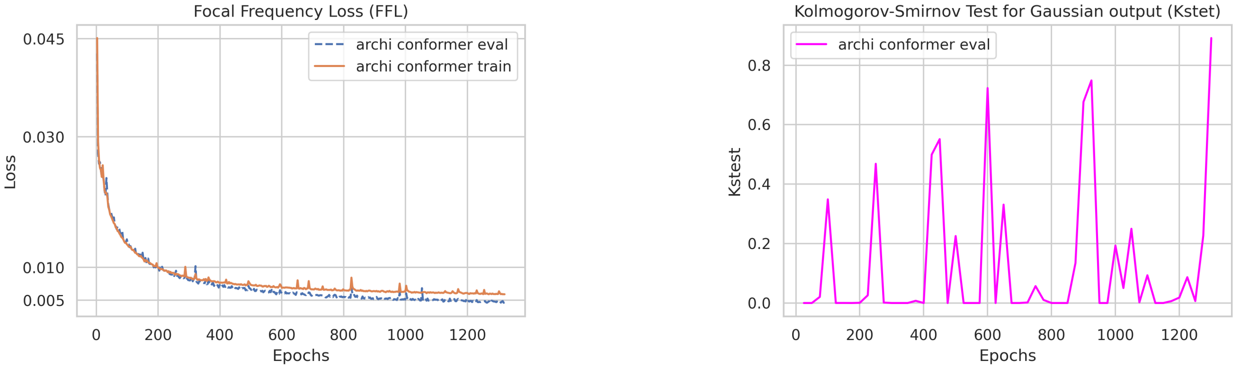
(a) (b) (c)

(d) (e) (f)

Figure 4.9: We present the unconditional generation of broadband data from $\mathbf{z}_y \sim \mathcal{N}(0, \mathbf{I}) : G_y(\mathbf{z})$. The purpose of presenting different values is to visually represent the range of signals generated by our model. This exploration contributes to a comprehensive understanding of the model's performance and capacity to capture variability in broadband signal generation.



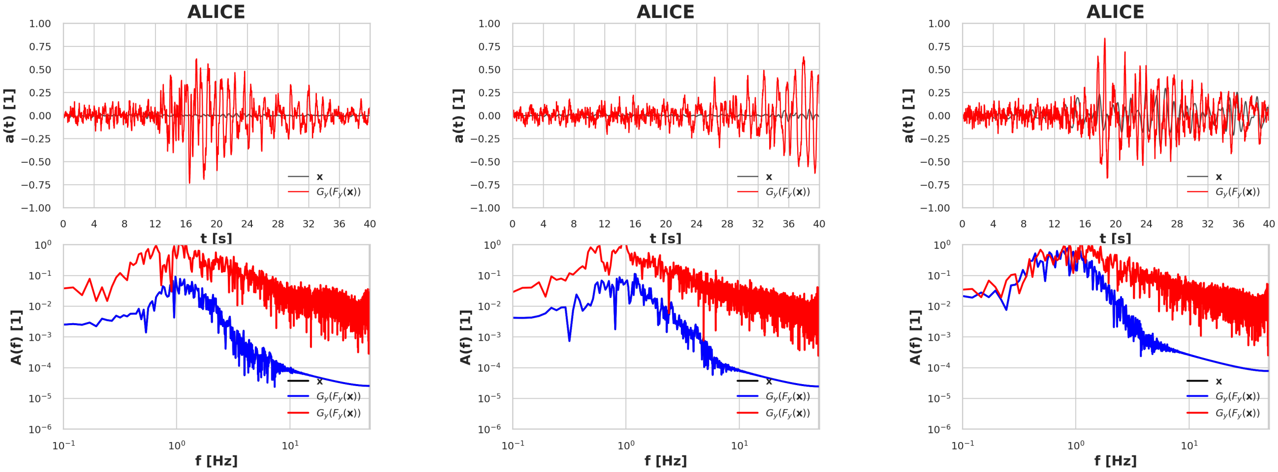
(a) Adversarial solution for ALI (b) Hyper Spherical Loss (HSL)



(c) Focal Frequency Loss (FFL) (d) Kolmogorov-Smirnov Test

Figure 4.10: Adversarial Solution Training. In (a) the Adversarial loss, ALI is represented, converging towards the optimal solution is $-\log(4) \approx 1.386$ the loss remains close to the solution due to TTUR dynamic training. In (b), we present the HSL of reconstruction quality; the optimal solution is getting close to 0, which is the expected theoretical result. In (c), the FFL loss is shown. We found a small overfitting of the loss; the theoretical optimal solution is 0. In (d), We present the *Kolmogorov-Smirnov* test to evaluate the quality of the targeted distribution, $F_y(\mathbf{y})$ that should be Gaussian. The ordinate value is the probability of the distribution being a Gaussian distribution.

(vi) **Robustness**
Finally, the architecture under development demonstrates an ability to identify patterns necessary to render broadband data (refer to Figure 4.11). The nuanced distinctions in the tweaked signal representation showcase the network's capacity to capture essential features required for broadband signal generation.



(a) (b) (c)

Figure 4.11: Generation of broadband signal, $G_y(F_y(\mathbf{x}))$, in red from filtered signal in blue, $\mathbf{y}$.

4.4 Super-resolved physics-based simulations

A physics-based numerical simulation renders low-frequency signals with neither explicit nor unique broad-band counterpart: the super-resolution problem is ill-posed by definition. However, in the framework of seismic risk assessment, a probabilistic approach is adopted, for which plausible realizations of seismic scenarios are taken into account to consider the median prediction and some uncertainty margins drawn around it. This is the standard way modern earthquake engineering deals with epistemic and aleatory uncertainties, naturally present in recorded seismograms. However, few empirical observations are available for each earthquake occurrence and at each site: earthquake catalogs have been consistently recorded since the seventies when analogical and digital seismographs started to be massively deployed worldwide. Therefore, statistical estimators drawn from recorded data encompass too high uncertainty. This is the case of Ground Motion Prediction Equations (GMPEs), *i.e.*, analytical ground motion models under equations predicting an earthquake intensity measure based on coefficients identified through non-linear regression on proxies and earthquake characteristics.
On the other hand, physics-based numerical simulations are deterministic realizations of site-specific seismological and geological conditions. Any physics-based simulation is associated with considerable computational burden, posing challenges in constructing statistically consistent estimators, such as routinely done with empirical observations. Therefore, the strategy presented hereafter aims to enhance the physics-based deterministic realizations with the natural broad-band uncertainty of natural records.

4.4.1 Model Architecture

The auto-encoder architecture for physics-based simulations is virtually identical to broadband signals. The only difference is the size of the latent space, which is reduced from 256 to 128. The physics-based signals are less complex than broadband signals, so we estimated that half of the parameters should suffice. We call F_x and G_x the physics-based simulations encoder and decoder, respectively. In addition, the encoder architecture mirrors the decoder architecture and closely resembles that of the broadband solution, see Figure 4.4, where we have examined different parts of it. We conserve the Conformer, the residual block, and how we pass from the data distribution space to the latent distributions. Concerning the discriminator, the only change is adapting the architecture for the latent space. Our style includes two branches as illustrated in Figure 4.5, let us called those branches D_{s_x} and $D_{s_{z_x}}$. D_{s_x} extracts relevant features of $\mathbf{x}$, and the other one, $D_{s_{z_x}}$, extracts information from the latent space, $\mathbf{z}_x$. After the two outputs are concatenated for a final extraction, the last value provides values that will be fed to the BCEWithLogit loss. Figure 4.5 illustrates the design of the architectures. We call the discriminator D_{xz}, the pair of joint distribution analyzed are : $(\mathbf{x}, F_x\mathbf{x})$ and $(G_x(\mathbf{z}_x), \mathbf{z}_x)$. This configuration enables the discriminator to effectively assess the relationship between the input $\mathbf{x}$ and its corresponding latent space representation $\mathbf{z}_x$ in the context of our joint optimization objective.

4.4.2 Experimental Design

The signals in the database are normalized by the broadband signal's corresponding peak ground acceleration (PGA). They are then subjected to a 1 Hz filtering process. However, there is a significant problem with this technique. The amplitude of the physics-based signal is reduced to values below 0.01. These low amplitudes could hinder the convergence of the training process.

We implement a re-scaling strategy on the physics-based data to surmount this challenge before commencing the training phase. By performing data re-scaling, we harmonize the amplitudes to a more suitable range, thus facilitating improved convergence during training. Once trained with re-scaled physics-based data, a subsequent training phase follows, where we undertake further training using the non-scaled signals, culminating in finalizing the entire process. In our experimentation, we evaluate the performance of our architecture using both scaled and unscaled physics-based data during the neural network training process. Specifically, we conduct testing before applying our auto-encoder and subsequently after by incorporating an adversarial loss. Before applying the auto-encoder, we assess the neural network's ability to process scaled and unscaled physics-based data during the testing phase. This step allows us to gauge the network's initial performance and ascertain differences in response to the two data types.

4.4.3 Results

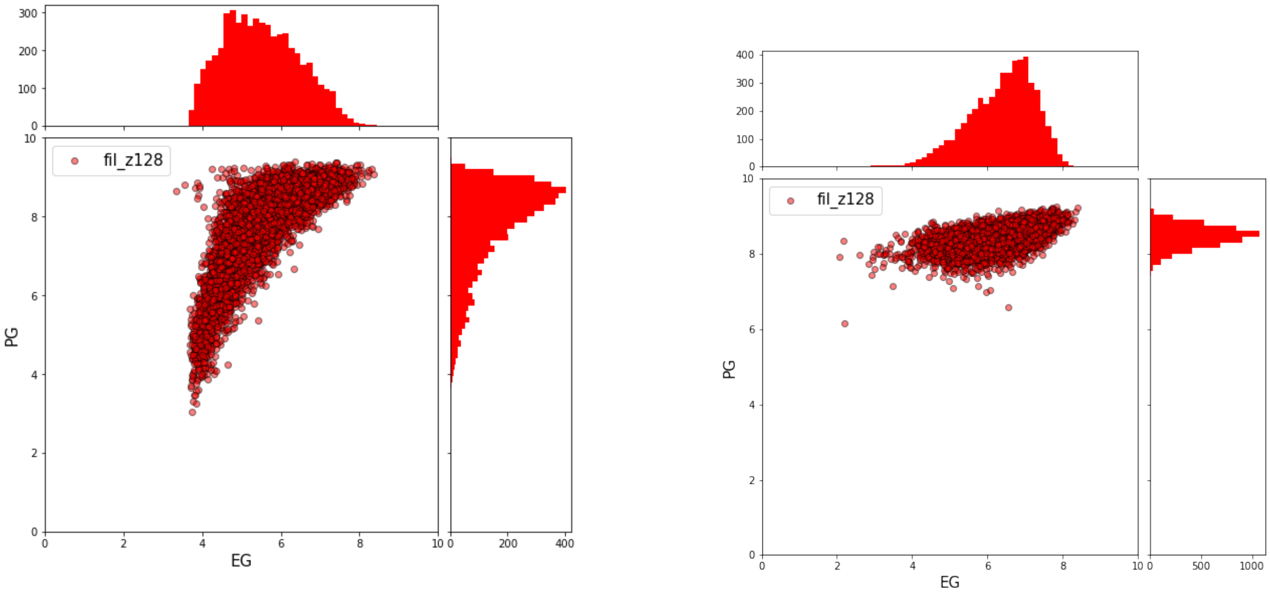
(i) **Comparison with the previous strategy**

Compared to Gatti et al., 2020 [21], who solely focused on the reconstruction of the broad-band signal based on the physics-based enhancement through SeismoALICE, we also focused hereafter on the unconditional generation of realistic broadband signals, using only the decoder, fed with Gaussian noise and precisely tailored to match the characteristic of the physics-based simulations data set.

(ii) **Data augmentation**

Moreover, we improve the training process by augmenting the size of the dataset and by incorporating an adversarial loss. By leveraging the adversarial loss, we guide the Generators toward generating output that exhibits increased robustness and closer adherence to the underlying data distribution. This process helps to improve the overall generalization and robustness of the model.

(iii) **Performance comparison with G.O.F**
The performance of the model is evaluated through the G.O.F. produced. These are shown in Figure 4.12. Figure 4.13 illustrates the quality of the reconstruction. Different accelerograms represent how the network could compress the information on a vector of 128 components. In the frequency domain, we note that the neural network can introduce artificial noise for frequencies above 10 Hz, which we should not observe. This anomaly persists despite the various advances and architectural tests carried out to date.

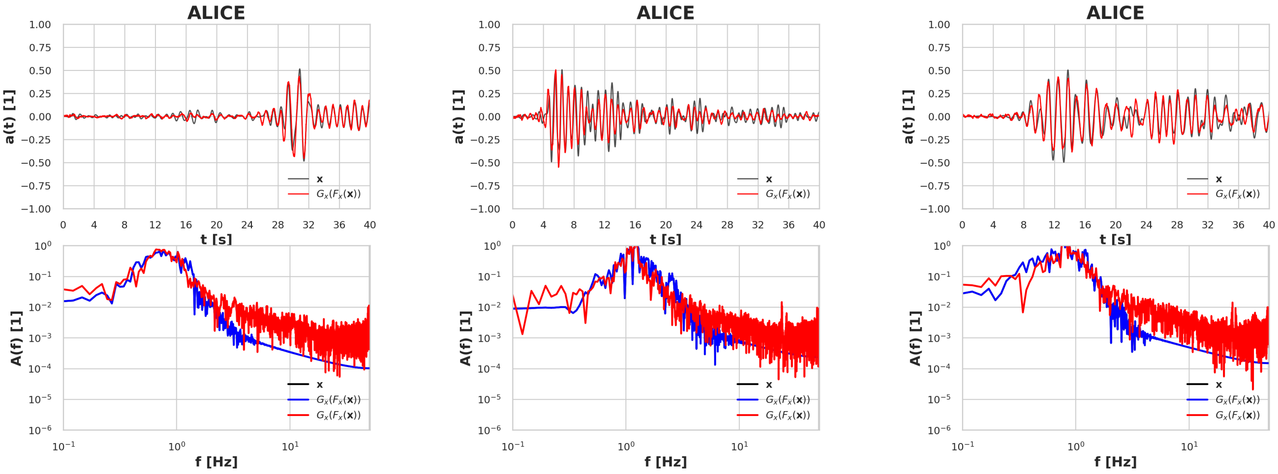


(a) List of Goodness of fit for the Autoencoder of the filtered signal for the scaled physic-based signal.

(b) Goodness of fit for the signal reconstructed after being trained with adversarial loss.

Figure 4.12: Goodness of Fit for the filtered signal. The subset of data comprehends 30000 signals. We could see how the adversarial loss significantly improves signal reconstruction quality.

(iv) **Generation from Gaussian and Latent space Analysis**
Figure 4.13 presents examples of different outputs from Gaussian noise, $\mathbf{z}_x$. The outputs produced serve as representative samples of the database. To better understand the arrangement of the data, we group the $F_x(\mathbf{x})$ values obtained from the test set and use PCA, t-SNE and uMap in 2D and 3D for a better observation of the main directions. No clustering was observed. See Figure 4.15. Each point represents a signal. The fact that using different methods leads to the same aspect of the data's reduced dimension clearly expresses the representation's compacity. In addition to that the Normal distribution also highlights the independence of each drawing. These aspects will be strongly used in Section §4.6.1 to transfer a data type to another.



(a) (b) (c)

Figure 4.13: We present the result for reconstructing some filtered signal.

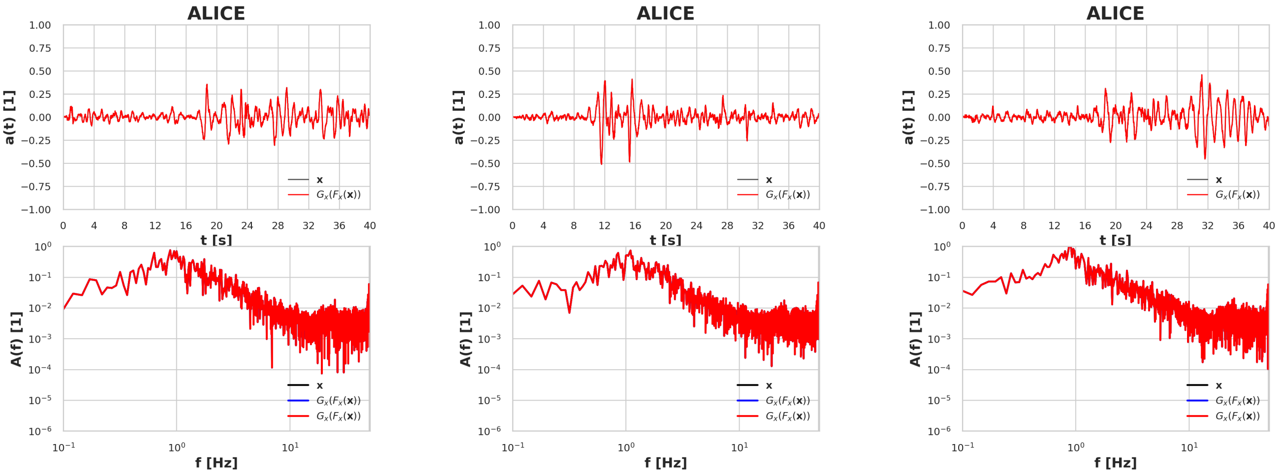
4.5 Possible solution: shared latent space

4.5.1 Motivation

There is an interdependence between physics-based simulation signals and broadband signals. We want to exchange information between them through latent space. We propose that the two data types share the same latent space to achieve this. We assume there will be two distinct representations: one for low-frequency components, *i.e.*, signals whose frequency content does not exceed 1Hz, and another space for signals with a higher frequency content, as observed in the frequency domain.

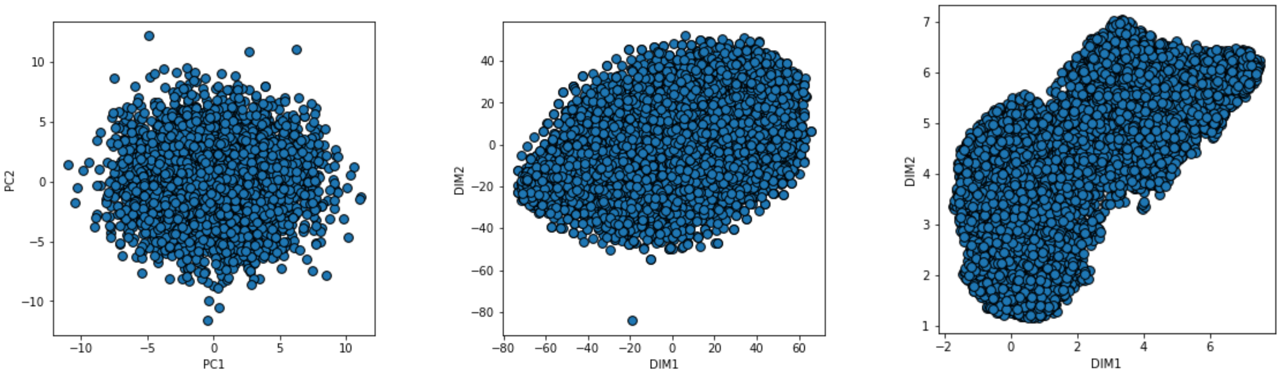
4.5.2 Methodology

This study builds on the shared latent space and contributes to viewing how the encoder could interpret the data coming from different sources in the same latent space. Much of the recent

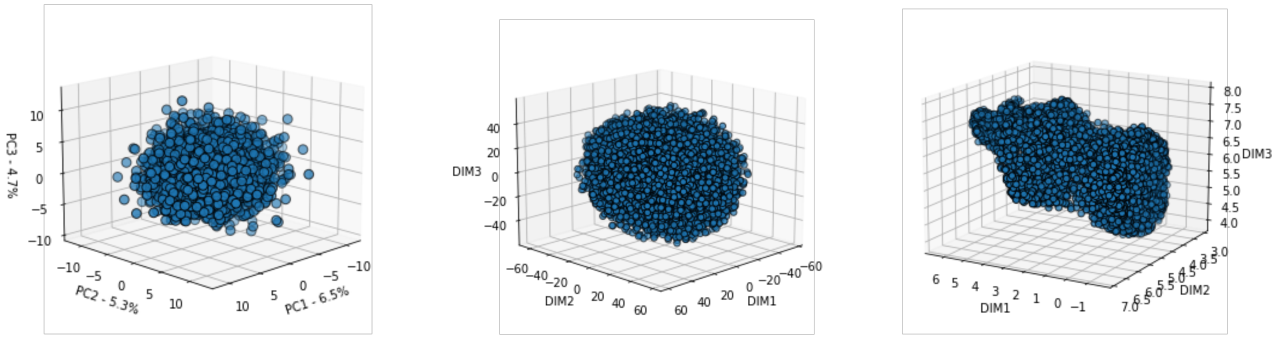


(a) (b) (c)

Figure 4.14: We present the result for the generation from the Gaussian distributions. $G_x(\mathbf{x})$



(a) 2D PCA (b) 2D t-SNE (c) 2D uMAP



(d) 3D PCA (e) 3D t-SNE (f) 3D uMap

Figure 4.15: We present the PCA, the t-SNE and the uMAP for the the Gaussian distributions of the broadband signal. $F_y(\mathbf{x})$

literature has overlooked this for time-histories. We encode the physics-based data and use the decoder to see if this could reproduce the corresponding broadband and vice-versa, according to the following scheme:

$$\mathbf{x} \xrightarrow[F_{xy}]{} \mathbf{z}_x \xrightarrow[G_y]{} \hat{\mathbf{y}} \tag{4.27}$$

$$\mathbf{y} \xrightarrow[F_{xy}]{} \mathbf{z}_y \xrightarrow[G_x]{} \hat{\mathbf{x}} \tag{4.28}$$

The hybrid approach is summarized in Figure 4.16. We introduce a new discriminator, D_{xy}, to force the hybrid generation to match the wanted values. This latter will have the task to distinguish between the pair of $(\mathbf{x}, \hat{\mathbf{y}})$ and $(\hat{\mathbf{x}}, \mathbf{y})$. So the cost function reads:

$$\mathcal{L}^{\mathrm{xy}}_{\mathrm{Hybrid}} = \mathbb{E}[\log D_{xy}(\mathbf{x}, \hat{\mathbf{y}})] + \mathbb{E}[\log(1 - D_{xy}(\hat{\mathbf{x}}, \mathbf{y}))] \tag{4.29}$$

$$\begin{aligned}
\mathcal{L}^{\mathrm{y}}_{\mathrm{ALI}} = &\quad \mathbb{E}[\log D_{yz}(\mathbf{y}, F_{xy}(\mathbf{y})] + \mathbb{E}[\log(1 - D_{xz}(G_y(\mathbf{z}_y), \mathbf{z}_y)] \\
\mathcal{L}^{\mathrm{x}}_{\mathrm{ALI}} = &\quad \mathbb{E}[\log D_{xz}(\mathbf{x}, F_{xy}(\mathbf{x})] + \mathbb{E}[\log(1 - D_{xz}(G_x(\mathbf{z}_x), \mathbf{z}_x)] \\
\mathcal{L} = &\quad \mathcal{L}^{\mathrm{x}}_{\mathrm{ALI}} + \mathcal{L}^{\mathrm{y}}_{\mathrm{ALI}} + \mathcal{L}^{\mathrm{xy}}_{\mathrm{Hybrid}} + \mathcal{L}^{\mathrm{x}}_{\mathrm{rec}} + \mathcal{L}^{\mathrm{y}}_{\mathrm{rec}} + \mathcal{L}^{\mathrm{x}}_{\mathrm{rechyb}} + \mathcal{L}^{\mathrm{y}}_{\mathrm{rechyb}}
\end{aligned} \tag{4.30}$$

In the previous expression, the different loss functions can be further unravelled as follows:

$$\begin{aligned}
\mathcal{L}^{\mathrm{x}}_{\mathrm{rechyb}} = &\quad \mathrm{FFL}(\mathbf{x}, G_x(F_{xy}(\mathbf{y})) + \mathrm{HSL}(\mathbf{x}, G_x(F_{xy}(\mathbf{y})) \\
\mathcal{L}^{\mathrm{y}}_{\mathrm{rechyb}} = &\quad \mathrm{FFL}(\mathbf{y}, G_y(F_{xy}(\mathbf{x})) + \mathrm{HSL}(y, G_y(F_{xy}(\mathbf{x}))
\end{aligned} \tag{4.31}$$

The development roadmap followed several subsequent steps:

(I) Reconstruction of recorded ground motion $\mathbf{y}$ as $G_y(F_{xy}(\mathbf{y}))$ (Figure 4.20)

(II) Reconstruction of physics-based simulation $\mathbf{x}$ as $G_x(F_{xy}(\mathbf{x})$ (Figure 4.20)

(III) Hybrid transformation from physics-based simulation to recorded ground motion via $\hat{\mathbf{y}} = G_y(F_{xy}(\mathbf{x}))$ (Figure 4.23)

(IV) Hybrid transformation form recorded ground motion to physics-based simulation via $\hat{\mathbf{x}} = G_x(F_{xy}(\mathbf{y}))$ (Figure 4.24)

(V) Unconditional generation of recorded ground motion $G_y(\mathbf{z}_y)$ (Figure 4.25)

(VI) Unconditional generation of physics-based simulation $G_x(\mathbf{z}_x)$(Figure 4.26)

4.5.3 Phase Alignment

In Figure 4.18, we notice that the previous technique alters the natural alignment of S-wave and P-wave in the hybrid data. We use a neural network designed for phase picking to force this aspect into the generative process. This latter serves as further constraint, added as ℓ^2 penalty loss. The pre-trained EQTransformer conceived by Mousavi *et al.*, 2020 [46] was adopted to this end. The architecture is designed for the signal length of 6000 time steps. Since signals of 4096 time-steps feature our dataset, the latter were zero-padded. The extra penalty loss reads:

$$\mathcal{L}_{EQT} = \|P - EQ_P(\hat{\mathbf{y}})\|_2 + \|S - EQ_S(\hat{\mathbf{y}})\|_2 \tag{4.32}$$

where P and S are the real P- and S-wave arrival times of the real signal, $\mathbf{y}$, whereas EQ_P and EQ_S are the corresponding predictions obtained with EQTransformer. A penalty factor λ_{EQ} is adopted. Adding this penalty loss corrects the misalignment of P and S-waves of hybrid generated data from physics-based inputs $\mathbf{x}$. The loss of the phase alignment could be observed in Figure 4.21.

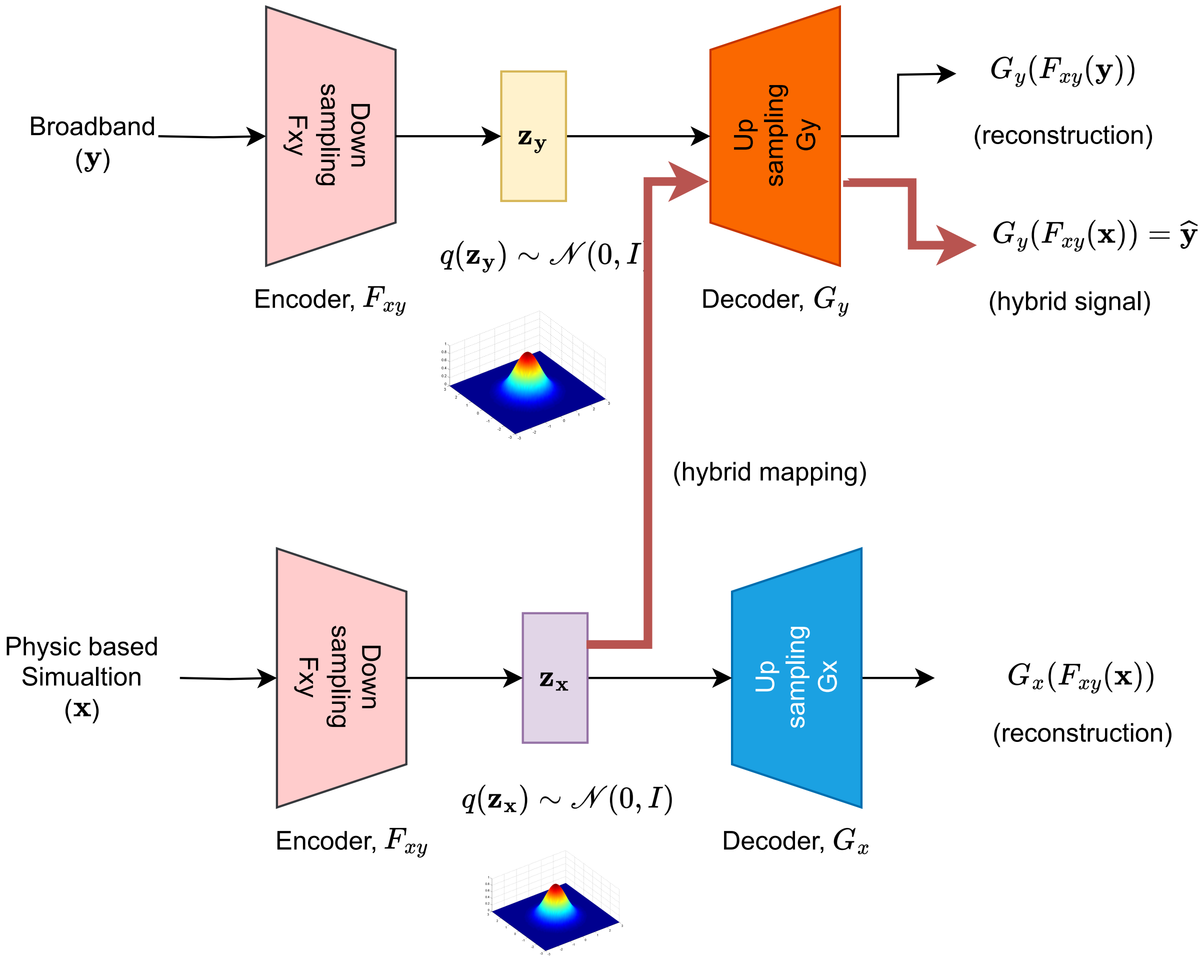


Figure 4.16: Sum up of the architecture of the shared latent space. The hybrid representation is obtained as $G_y(F_{xy}(\mathbf{x}))$.Even though we have duplicated the encoder on the diagram, it is the same encoder.

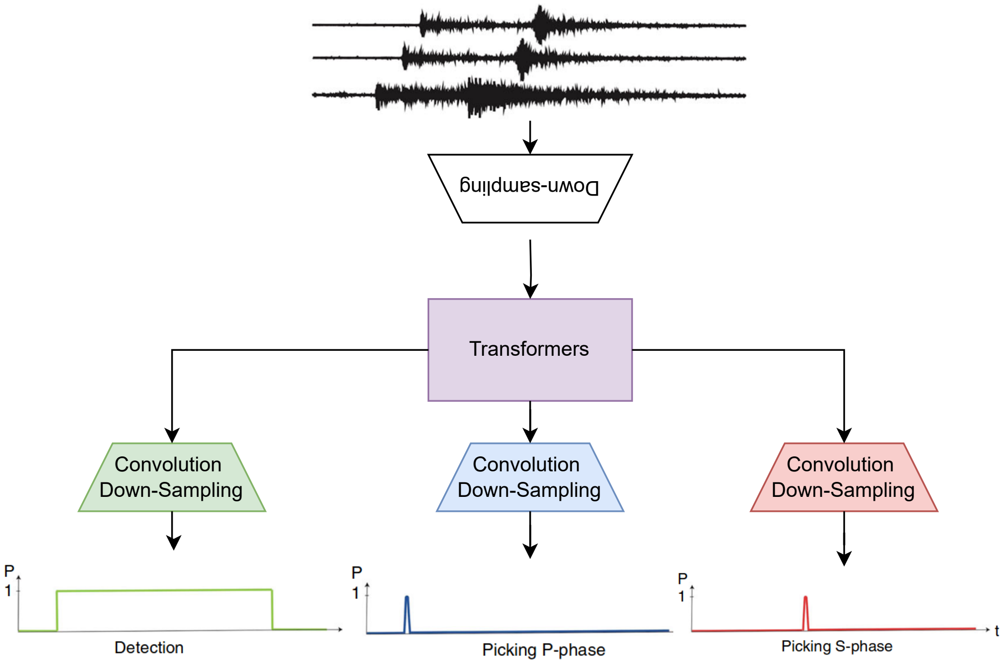


Figure 4.17: EQTransformer architecture by Mousavi was conceived to detect Noise, P-wave, and S-wave arrival. Its output (Noise, P position, and S position) is used to help the output stay close to the targeted ground motion reconstructions of the data. Source [51]

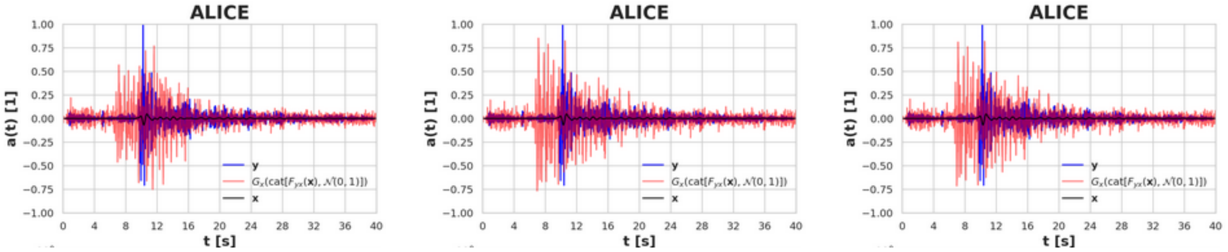


Figure 4.18: Illustration of non-alignment of phase for a seismogram. We solve this issue with a loss performed through the EQTransformers (Equation 4.32).

4.5.4 Total training Loss

The holistic training process integrates three key components: the adversarial loss, the reconstruction loss, and the EQTransformer loss, serving as features extractor. Each component contributes to the overall training loss; specific weighting parameters regulate their revealed importance. In our case we use as parameters $\lambda_{\text{adv}} = 1$, $\lambda_{\text{rec}} = 10$ and $\lambda_{\text{EQ}} = 0.1$. These values are a trade-off that stabilizes the influence of the different parts of the whole training process. The evolution of the losses and metrics can be seen in Figure 4.21. We comprehensively understand how each component contributes by visualizing the entire spectrum of losses and metrics. Because all the elements have been separately studied before being combined, we can assess that each could not be removed without endangering the stability of the training process.

$$\begin{aligned}\mathcal{L} = {} & \lambda_{\text{adv}}(\mathcal{L}^{\text{x}}_{\text{ALI}} + \mathcal{L}^{\text{y}}_{\text{Hybrid}} + \mathcal{L}^{\text{xy}}_{\text{ALI}}) \\ & + \lambda_{\text{rec}}(\mathcal{L}^{\text{x}}_{\text{rec}} + \mathcal{L}^{\text{y}}_{\text{rec}} + \mathcal{L}^{\text{x}}_{\text{rechyb}} + \mathcal{L}^{\text{y}}_{\text{rechyb}}) \\ & + \lambda_{\text{EQ}}\mathcal{L}^{\text{EQ}}\end{aligned} \tag{4.33}$$

4.5.5 Unique encoder and new joint discriminators design

A unified solution has limits, even when adding the EQTransformer to condition the arrival times better. To further improve the super-resolution mapping, the encoders F_y, F_x were merged into a unique encoder with two branches F_{xy}, whereas the two decoders (G_y, G_x) were kept apart. This merger enables the encoder to map both $\mathbf{x}$ and $\mathbf{y}$ into a Gaussian manifold of two separate supplementary ones.

We conserve the architecture discriminators D_{xz}, D_{yz}, and we introduce a new discriminator D_{xy}. The particularity of D_{xy} is now the fact that it shares weights with D_{yz} and D_{xz} discriminator. The illustration of the Incorporated part of the discriminator of D_{xz} and D_{yz} is presented in Figure 4.19. The Equation modeling is presented below:

$$D_{xy}(\mathbf{x}, \mathbf{y}) = D_{xy}(D_{s_x}(\mathbf{x}), D_{s_y}(\mathbf{x})) \tag{4.34}$$

We do not recreate those discriminator branches from scratch, but we reuse them to complete the discriminator D_{xy}. In this sense, all the discriminators are aware of the improvement in communicating with the encoder and the generators by introducing complexity. Also, reusing the same part to form D_{xy} stabilizes the training of the generators and reduces the memory placed in the GPU, allowing faster training. To focus on discriminating hybrid mappings in the unified domain, we use Equation 4.29.

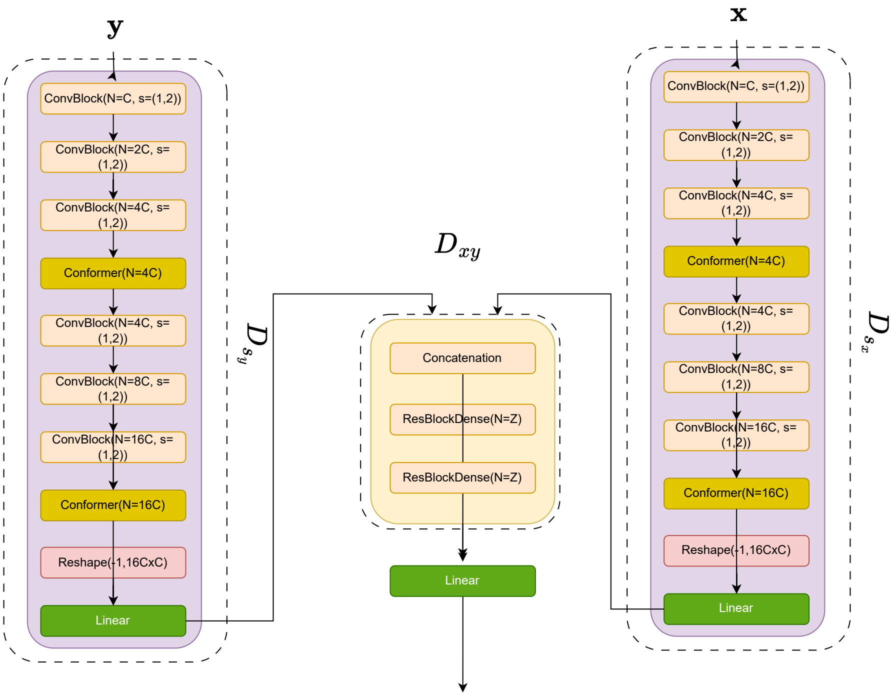


Figure 4.19: We present the architecture of the joint discriminator over the pair $(\mathbf{x}, \hat{\mathbf{y}})$ and $(\hat{\mathbf{x}}, \mathbf{y})$. We do not recreate, D_{s_y} and D_{s_x} respectively incorporated on the design architecture of D_{yz} and D_{xz}. These weights are directly used in the D_{xy} architecture. In this sense, the diverse architecture also gives a view of the hybrid signal and the original one.

4.5.6 Experiments

(i) **Case Application for Phase Alignment**
The training of the Model without the EQTransformers is illustrated in Figure 4.18. Then, we used the abilities of the EQTransformers. Initially, the training of EQTransformer model was done under Cross Entropy loss, but for our ALICE training, this was not efficient. Subsequently, a switch to ℓ^2 loss was made, offering a more tailored solution for our intended phase picking alignment. The coefficient chosen also plays a crucial role in stabilizing the training. Following experimentation, we established that among 10, 1, 0.5, and 0.1, 0.1 strikes a balance hybrid generation acceptable.

(ii) **Hyper Parameter Tuning**
Finally, as a hyperparameter, we conserve a learning rate of 4×10^{-4} for the generators (encoder and decoder) and a learning rate of 1×10^{-4} for the discriminators. As an optimizer, we use Adam.

4.5.7 Results

(i) **Performance evaluation with G.O.F**
Our preliminary investigation was conducted to ascertain the fulfillment of our objectives outlined in the methodology (Section 4.5.2). Network performance can be evaluated using GOFs. Figure 4.20, gather all the results for EG and PG to assess the quality of the reconstruction of the physics-based simulations signals, $G_x(F_{xy}(\mathbf{x}))$, the broadband signals, $G_y(F_{xy}(\mathbf{y}))$, and the hybrid broadband signals which come from the seismic signals, $G_y(F_{xy}(\mathbf{x}))$. Here, Objectives I, II, and III are satisfied. Compared to Pix2Pix,

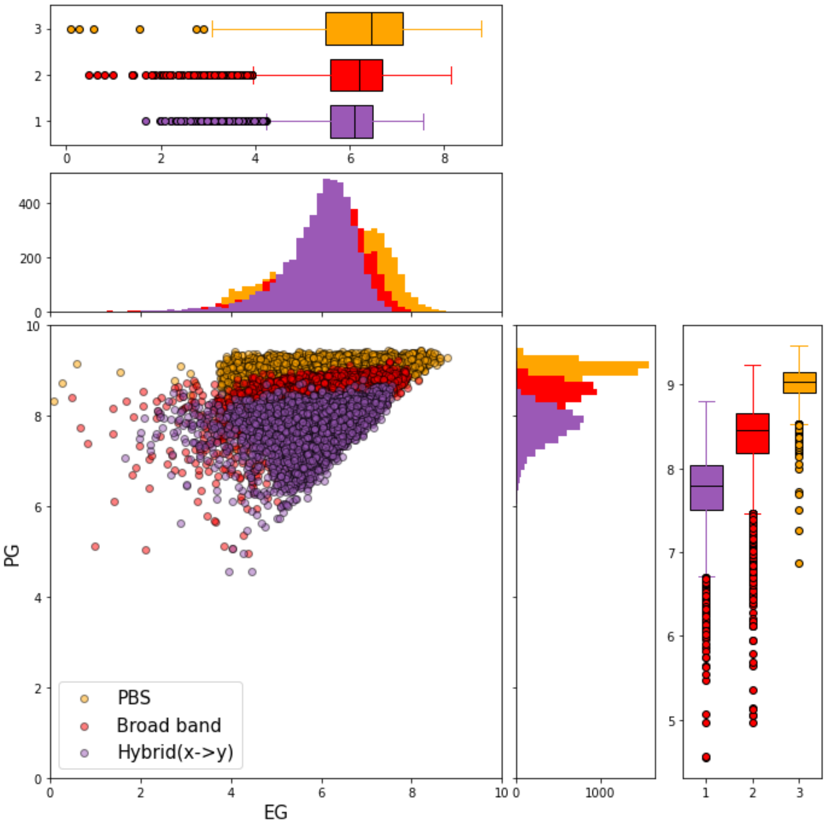


Figure 4.20: List of G.O.F for ALICE with shared latent space.

Figure 4.20 shows a lower reconstruction metric due to the lack of skip connections. Nevertheless, Pix2Pix is highly sensitive to noise and provides no robustness when we change the data type, drastically reducing its performance. In a sense, pix2pix learns to memorize the filter. However, the choice to ease unconditional generation from the latent space has some advantages. Those results highlight the feasibility of diverse cycling. We successful encode and decode both physics-based simulations ($\mathbf{x} \overset{F_{xy}}{\rightarrow} \mathbf{z}_x \overset{G_x}{\rightarrow} \tilde{\mathbf{x}}$) and recorded data ($\mathbf{y} \overset{F_{xy}}{\rightarrow} \mathbf{z}_y \overset{G_y}{\rightarrow} \tilde{\mathbf{y}}$). The unconditional generation process is also addressed, allowing us to generate broadband and filtered data from multivariate normal variables $\mathbf{z}$.

(ii) **Unconditional Generation**
Figure 4.25 and Figure 4.26 is the proof that the objectives V and VI of Section § 4.5.2 are satisfied. These figures showcase the quality of the generated data from the Gaussian noise, proving that our strategy has effectively mitigated the mode collapse. The diversity and realism of these generated samples demonstrate the model's ability to learn and represent the diverse patterns and characteristics of the original data distributions by the absence of repeated patterns.

(iii) **Latent Space Interpretability**
An interesting aspect resides in how the encoded data are structured. t-SNE and uMap plots in Figure 4.22 clearly shows two clusters, *i.e.* two distinct representations for $\mathbf{x}$ and $\mathbf{y}$ respectively.

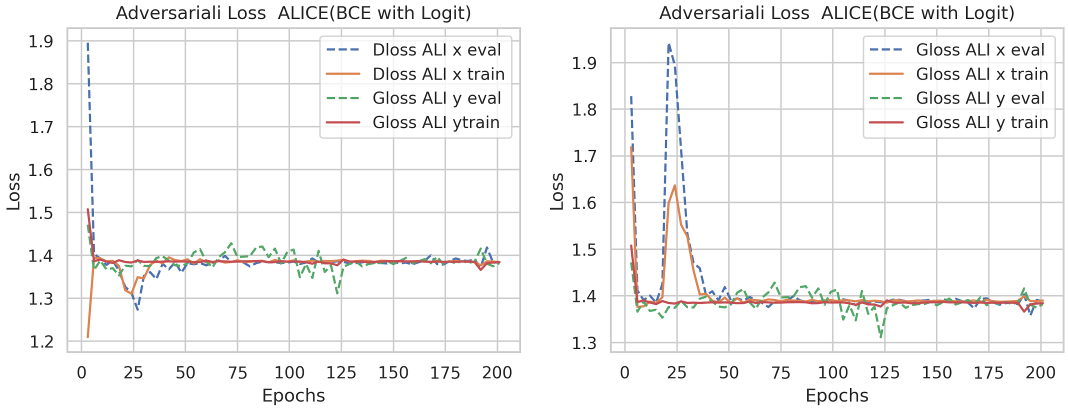
(iv) **Conditional Reconstruction**
Particular attention was devoted to conditional generation, where we observed the synthesis of synthetic data closely resembling the targeted broadband signal $\mathbf{y}$, based on the encoding of the filtered version $\mathbf{x}$ (Figure 4.23) and also for the reverse path (Figure 4.24). This multifaceted analysis provides a robust understanding of the capabilities and performance of our proposed methodology in the context of data representation and synthesis. The Objectives III and IV of Section § 4.5.2 are satisfied.

4.5.8 Limitations

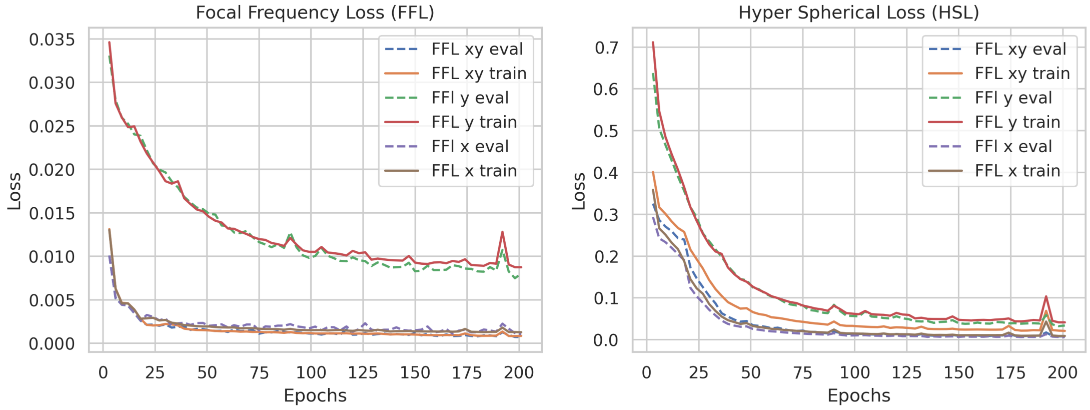
(i) **Assessing Degradation Despite Advances in Broadband and Filtered Solutions**
Despite the advancements made by various iterations, in our approach, we observed a degradation in the quality of reconstruction when compared to the solution presented in Sections § 4.2 and § 4.4. We have experienced robust architecture, but even after parameter regularization, the decline in the reconstruction quality persists.

(ii) **Artifact aspect**
Furthermore, we noticed spurious artifacts in the generated samples. A detailed observation has revealed that the training of ALICE introduces higher frequency irrelevant amplitude, deviating from the intended values. This limitation is not problematic for our exploration task but proves that the generated signals are synthetic. A highly efficient discriminator by exploration could easily interpret and distinguish real from fake. Without regularization parameters, accomplishing the training would be practically impossible.

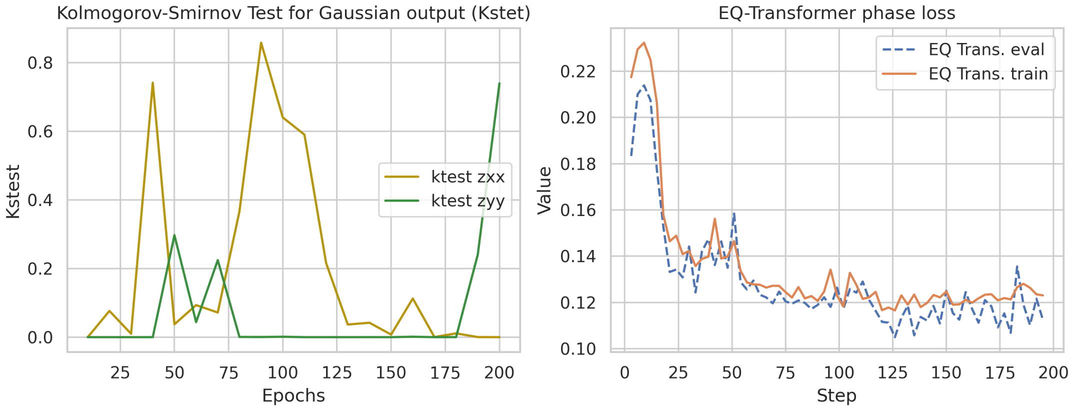
(iii) **Comparison with Pix2Pix**
Like Pix2Pix, our shared latent space enables broadband signal generation from the filtered data. Although the translation is feasible and provides an improved understanding



(a) Adversarial loss for the training of ALI. We present both the discriminator loss and the generator loss.

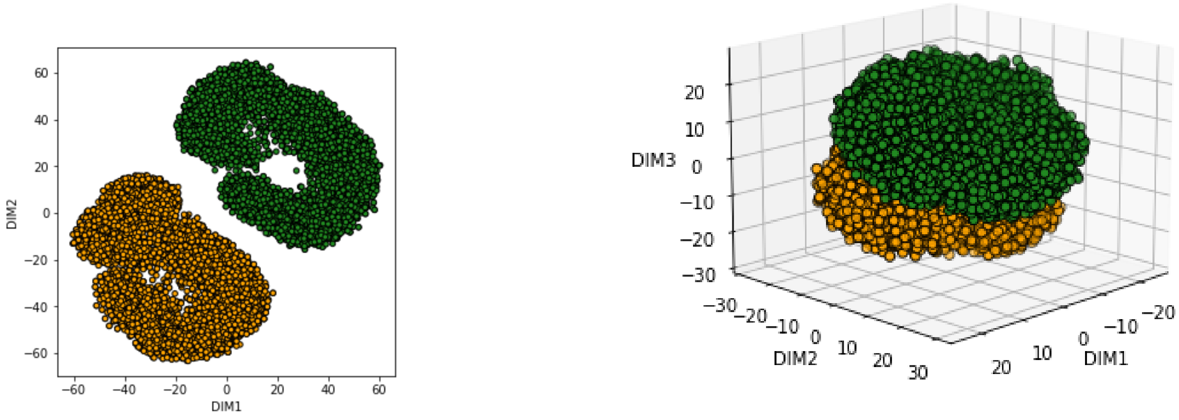


(b) Loss for reconstruction. We present the FFL and HSL for the reconstruction of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{x} \rightarrow \mathbf{y}$.



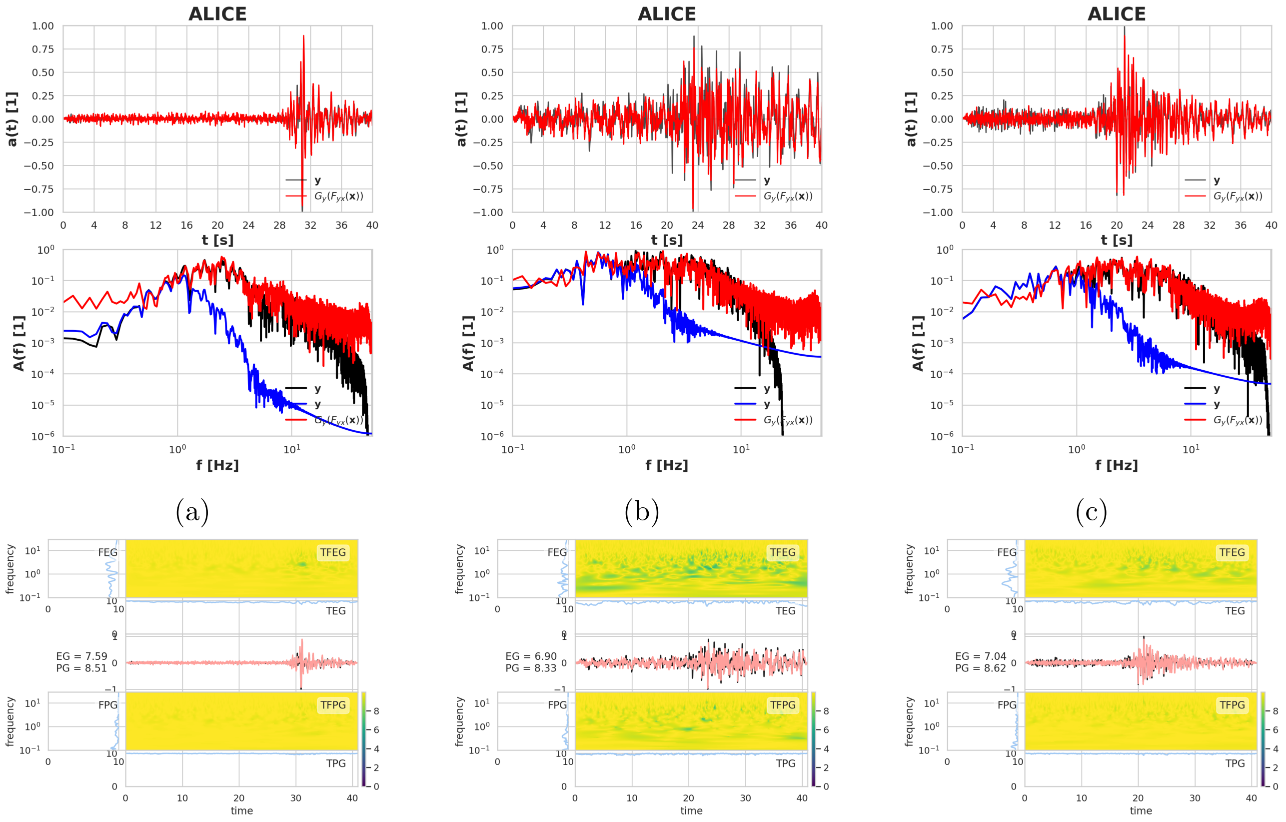
(c) Presentation of the Kolmogorov Test and for the phase picking with EQTransformer. The training takes 24h on 4 GPU A100, CUDA 11.7

Figure 4.21: Different losses for training ALICE unified strategy and shared latent space values.



(a) (b)

Figure 4.22: We present the result for the encoded representation of broadband and filtered signal t-SNE and uMap. $F_{xy}(\mathbf{x})$ (in green) and $F_{yx}(\mathbf{y})$ (in yellow). This is the proof of our previous assumption in Section § 4.5.2.

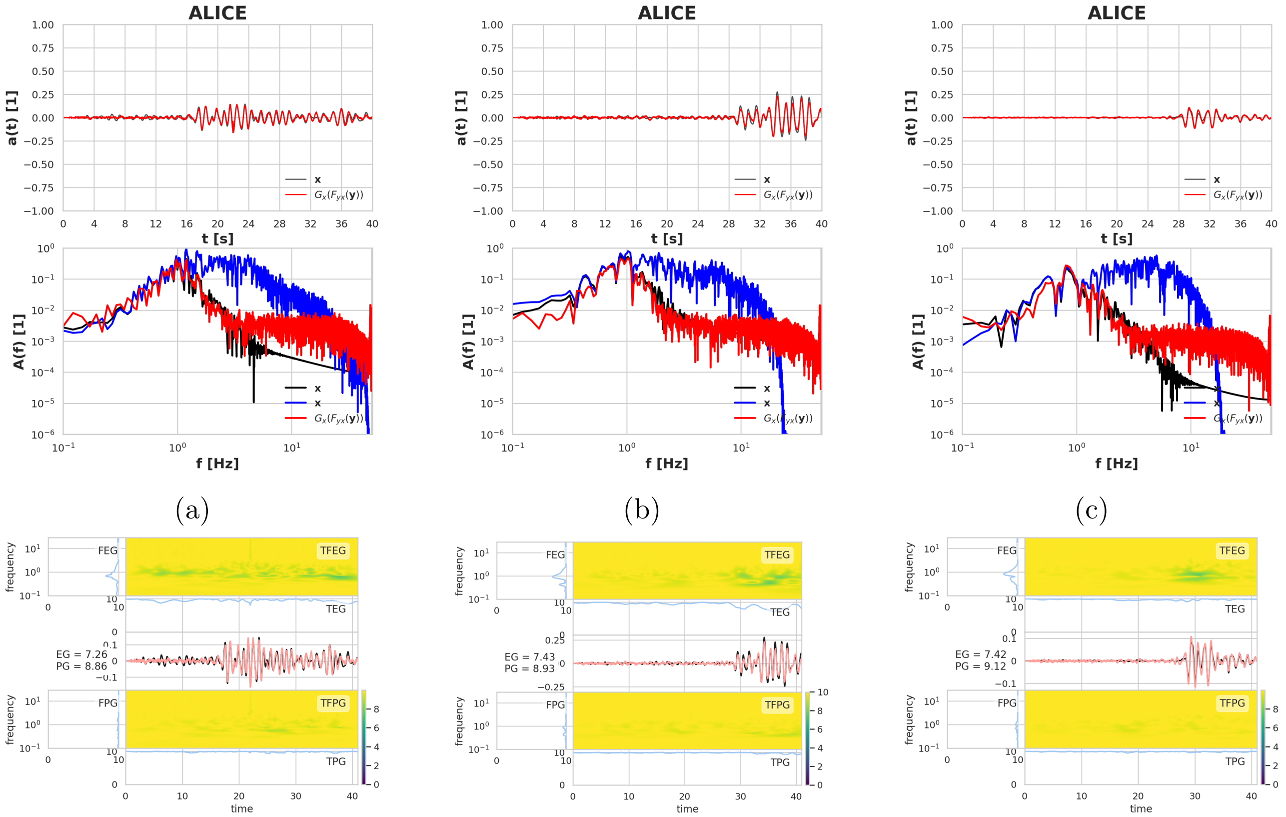


(a) (b) (c)

(d) (e) (f)

Figure 4.23: We present the result for the Hybrid generation broadband signal from the filtered signals: $G_y(F_{xy}(\mathbf{x}))$. This sis the proof of the objective III of Section § 4.5.2

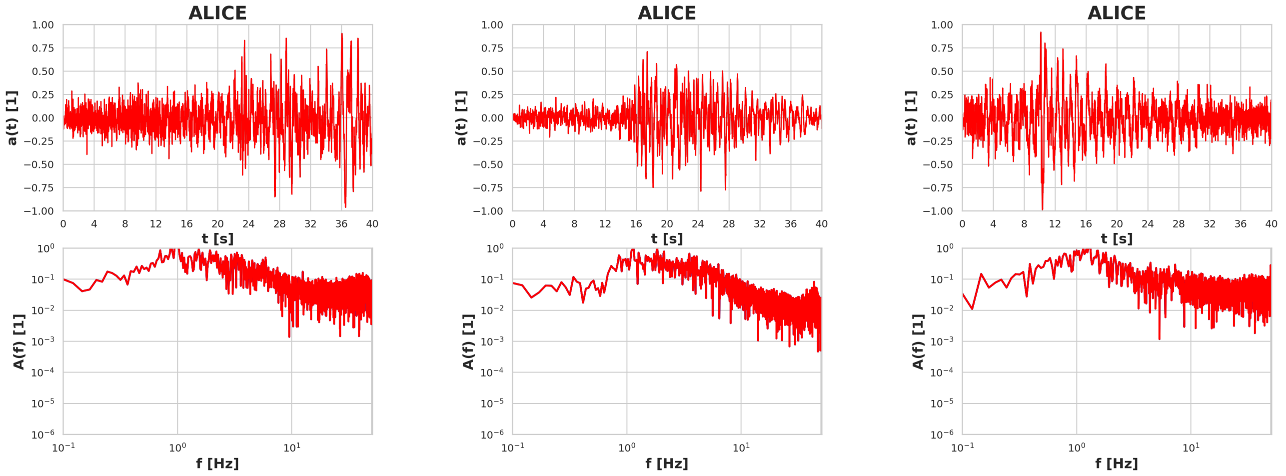
of the representation of latent information, and even though the translation is possible on both sides, one-to-many mapping is not possible. This limitation is acknowledged,



(a) (b) (c)

(d) (e) (f)

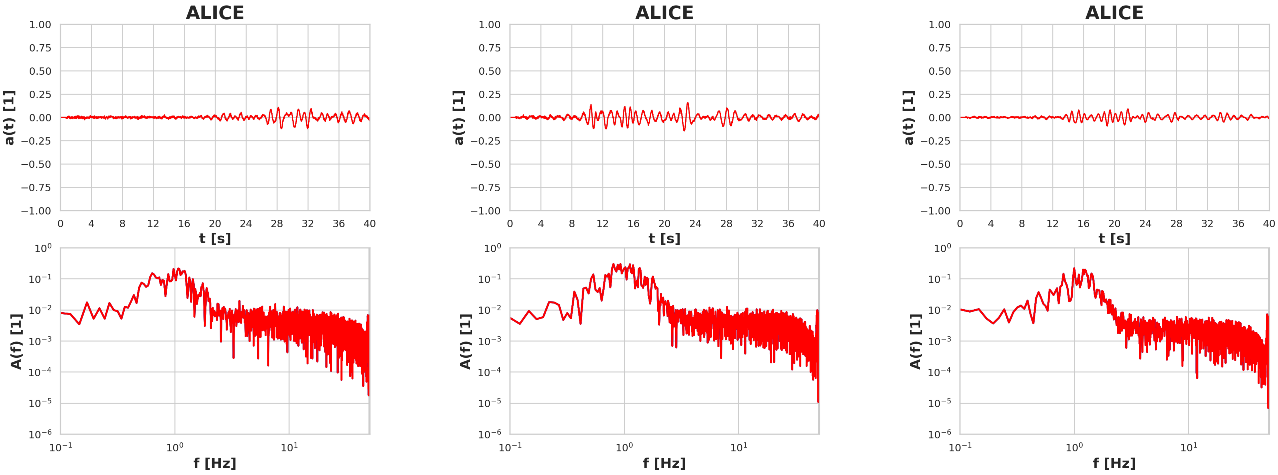
Figure 4.24: We present the result for the Hybrid generation broadband signal from the filtered signals: $G_x(F_{xy}(\mathbf{y}))$. This is the proof of the objective IV of the Section § 4.5.2.



(a) (b) (c)

Figure 4.25: We present the result for $G_y(\mathbf{z}_y)$. This is the proof of the objective V in the Section § 4.5.2

and the forthcoming Section 4.6 will address this challenge by introducing the factorized latent space, thus overcoming the current bottleneck and offering improved flexibility in handling complex civil engineering scenarios.



(a) (b) (c)

Figure 4.26: We present the result for $G_x(\mathbf{z}_x)$. This is the proof of the objective VI in the Section § 4.5.2.

4.6 Latent Space factorization

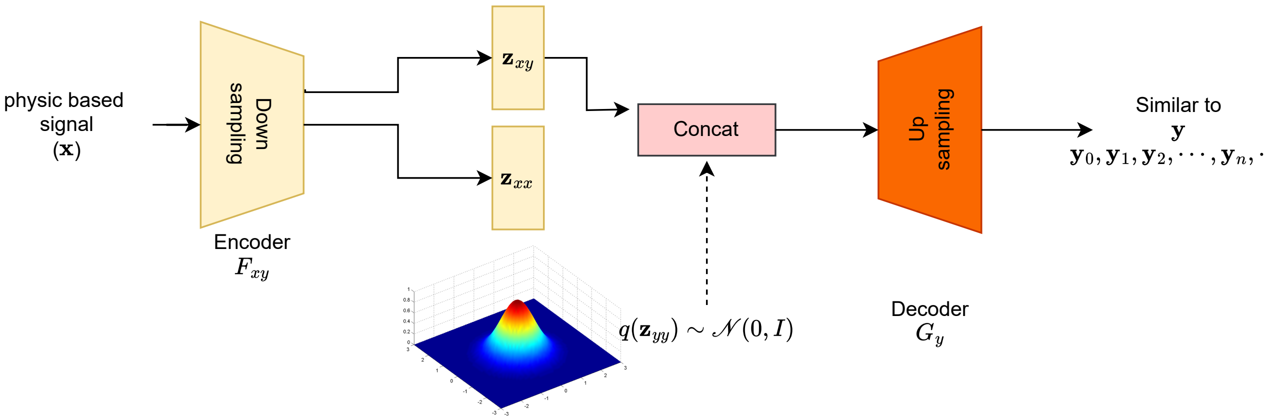


Figure 4.27: Inference for factorized latent space values. Hybrid generation with one-to-many mapping.

In our effort to refine the latent space representation, we acknowledge that the information presented in the Physics-Based data (or filtered data) is also inherently embedded in the recorded data. Our goal is to develop the representation of the latent that facilitates effective partitioning of the information. In light of what has been presented in the previous section, our methodology provides an explainable clustering of the latent representation. Based on this aspect, we intend to go beyond these limitations and condition the generation to perform a one-to-many mapping: one $\mathbf{x}$ and multiple $\mathbf{y}$ share the same portion of the spectrum corresponding to $\mathbf{x}$.

Rather than force one latent space per type of data ($\mathbf{x}$ or $\mathbf{y}$), we introduce two key components: a common latent space and a specific latent space operator. The common latent space, denoted as $\mathbf{z}_{xy}$ or $\mathbf{z}_{yx}$, whether it is coming from the input type $\mathbf{x}$ or the input of $\mathbf{y}$, is designed to encode the [0-1] Hz portion of the signal Fourier spectrum. In contrast, the specific latent part denoted $\mathbf{z}_{xx}$ or $\mathbf{z}_{yy}$ respectively, is tailored to capture the portion of the spectrum that is not shared by $\mathbf{x}$ and $\mathbf{y}$. This strategic partitioning is intended to ensure that the encoding of the physics-based signal $\mathbf{z}_{xx}$ becomes useless for hybrid generation. The $\mathbf{z}_{yy}$ component, on the contrary, specializes in the encoding of the peculiar features of $\mathbf{y}$, not shared with $\mathbf{x}$. Simultaneously, the recorded and physics-based signals share the same common features (the 0-1 Hz band) encoded into $\mathbf{z}_{xy} = \mathbf{z}_{yx}$ (see Figure 4.27).

4.6.1 Methodology

Our approach's motivation is to enhance the versatility of our latent space representation. Therefore, we have introduced this factorization of the latent space, allowing for a more nuanced and expanded one-to-many mapping. In the Equation 4.36 and Equation 4.35, we expose how we intend to achieve our goal. View in Figure 4.28 how we proceed to reconstruct Broadband and Physics-Based Simulation while considering the latent representations of each part of the data.

$$\mathbf{x} \xrightarrow[F_{xy}]{} \mathbf{z}_x = \left(F_{xy}(\mathbf{x})^{xy}, F_{xy}(\mathbf{x})^{xx}\right) \xrightarrow[\text{trunc}]{} \hat{\mathbf{z}}_{xy} \xrightarrow[G_x]{} \tilde{\mathbf{x}} \tag{4.35}$$

$$\mathbf{y} \xrightarrow[F_{xy}]{} \mathbf{z}_y = \left(F_{xy}(\mathbf{y})^{xy}, F_{xy}(\mathbf{y})^{yy}\right) \xrightarrow[\text{cat}]{} \hat{\mathbf{z}}_y \xrightarrow[G_y]{} \tilde{\mathbf{y}} \tag{4.36}$$

In this Equation, the symbol *cat* is for the concatenation of the two components of the latent space. The symbol *trunc* refers to extracting the standard part we extract for the complete latent space. This refers to a sub-vector. At the end of the training, the generators should consider the value of the latent space from $\mathbf{x}$ and the Gaussian noise and then transform it into a signal from the same database. Suppose the training is relevant by only replacing the common part with Gaussian noise and replacing the specific part with. In that case, the decoder should understand this is a simulated signal whose frequency is between [0 -1 Hz]. The specific part should give complementary information, such as a signal for which the frequency field is between [1 - 30] Hz.

$$\begin{aligned} F_{xy}(\mathbf{x}) &= (\mathbf{z}_{xy}, \mathbf{z}_{xx}) \\ F_{xy}(\mathbf{y}) &= (\mathbf{z}_{yx}, \mathbf{z}_{yy}) \end{aligned} \tag{4.37}$$

As shown in Equation 4.35, we have to pay attention to $\mathbf{z}_{xy}$ to train the adversarial discriminator for physics-based simulations. The following equation is the expression of the loss :

$$\mathcal{L}_{\text{ALI}}^{\text{x}} = \mathbb{E}[\log D_{xz}(\mathbf{x}, F_{xy}(\mathbf{x})^{xy}] + \mathbb{E}[\log(1 - D_{xz}(G_x(\mathbf{z}_x), \mathbf{z}_x)] \tag{4.38}$$

For the broadband signal, We consider the complete latent space with the following loss:

$$\mathcal{L}_{\text{ALI}}^{\text{y}} = \mathbb{E}[\log D_{yz}(\mathbf{y}, \text{cat}(F_{xy}(\mathbf{x})^{xx}, F_{xy}(\mathbf{y})^{yy})] + \mathbb{E}[\log(1 - D_{xz}(G_{\mathbf{y}}(\mathbf{z}_d), \mathbf{z}_d)] \tag{4.39}$$

For the hybrid discrimination, we choose using a loss that observes the pair of $(\mathbf{x}, \hat{\mathbf{y}})$ and the pair $(\hat{\mathbf{x}}, \mathbf{y})$ as we have done in the previous strategy. The expression of $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ is express as bellow :

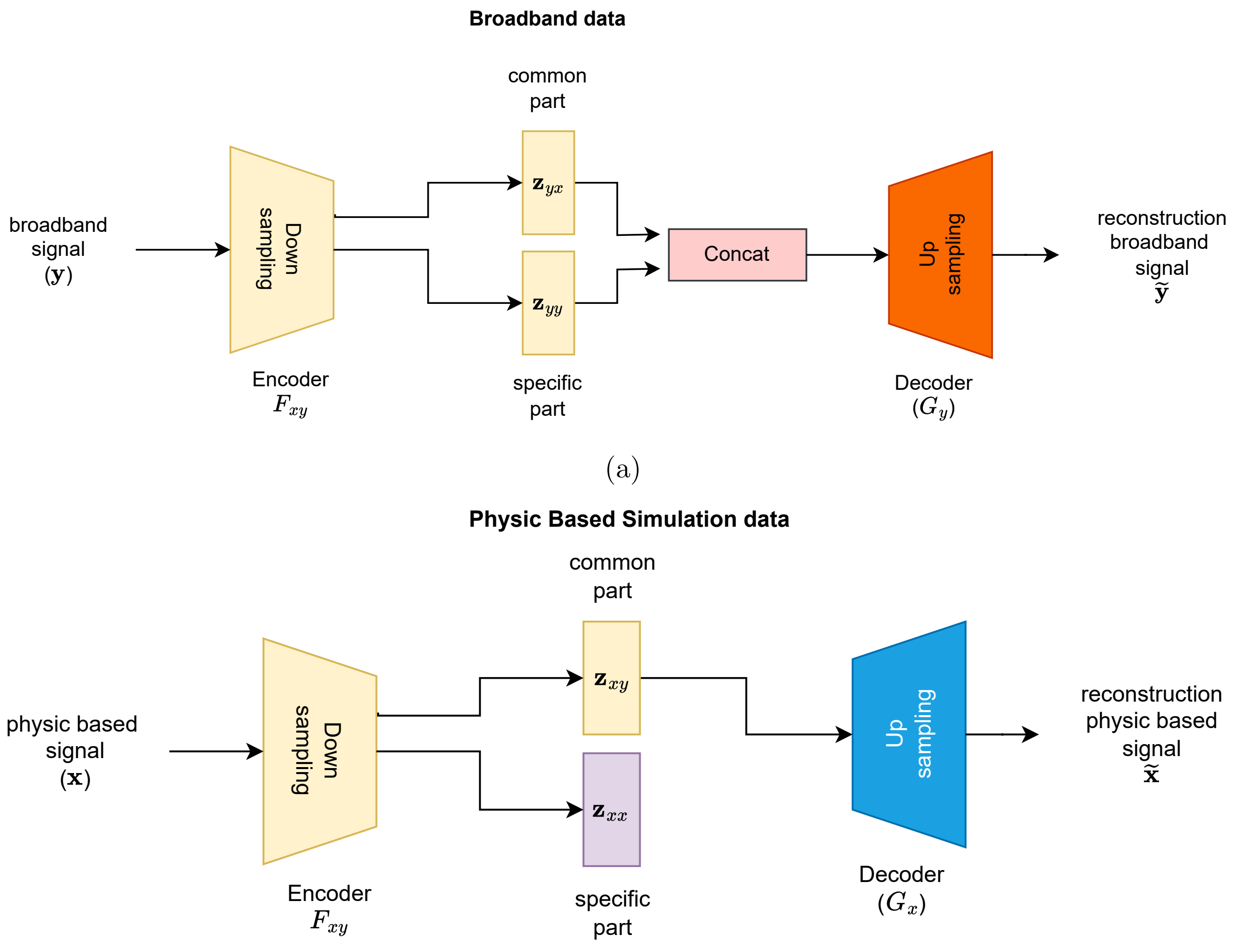
$$\mathbf{x} \xrightarrow[F_{xy}]{} \text{cat}(\mathbf{z}_{xx}, \mathcal{N}(0, I)) \xrightarrow[G_y]{} \hat{\mathbf{y}} \tag{4.40}$$

$$\mathbf{y} \xrightarrow[F_{xy}]{} \mathbf{z}_d \xrightarrow[\text{troncat}]{} \mathbf{z}_{xx} \xrightarrow[G_x]{} \hat{\mathbf{x}} \tag{4.41}$$

Then, the hybrid adversarial loss is presented before :

$$\mathcal{L}_{\text{Hybrid}}^{\text{xy}} = \mathbb{E}[\log D_{xy}(\boldsymbol{x}, \hat{\boldsymbol{y}})] + \mathbb{E}[\log(1 - D_{xy}(\hat{\boldsymbol{x}}, \boldsymbol{y}))] \tag{4.42}$$

Therefore, we want to satisfied :



(a)

(b)

Figure 4.28: Inference of Broadband and Physic-Based Simulation. We illustrated how the latent part is split into two parts. The $\mathbf{z}_{xx}$ is not used for the reconstruction of physics-based simulations, while the whole latent values are used for Broadband Signal ($\mathbf{z}_{yx}$ and $\mathbf{z}_{yy}$). The whole part of the latent follows a distribution $\mathcal{N}(0, \mathbf{I})$.

(I) Reconstruction of ground motion, $G_y(F^{yx}_{xy}(\mathbf{y}), F^{yy}_{xy}(\mathbf{y}))$;

(II) Reconstruction of physic based simulation, $G_x(F^{xy}_{xy}(\mathbf{x}))$;

(III) Force the coherence of common part, $F^{xy}_{xy}(\mathbf{x}) = \mathbf{z}_{xy} \sim F^{xy}_{xy}(\mathbf{y}) = z_{yx}$;

(IV) Force each part to follow Gaussian Distributions

(V) Generation of $G_y(\mathbf{z}_y)$;

(VI) Generation of $G_x(\mathbf{z}_x)$;

(VII) Generation of many hybrid output $G_y(\mathbf{x}, \mathcal{N}(0, \mathbf{I}))$

(VIII) Force the phase alignments for generation one to many

4.6.2 Total loss

By superposition of those losses, we finally obtained :

$$\begin{aligned} \mathcal{L} = \lambda_{\text{adv}} \left(\mathcal{L}^{\text{x}}_{\text{ALI}} + \mathcal{L}^{\text{y}}_{\text{ALI}} + \mathcal{L}^{\text{xy}}_{\text{Hybrid}} \right) \\ + \lambda_c (\mathcal{L}^{\text{x}}_{\text{rec}} + \mathcal{L}^{\text{y}}_{\text{rec}} + \mathcal{L}^{\text{x}}_{\text{rechyb}}) + \lambda_{\text{EQ}} \mathcal{L}_{EQT} \end{aligned} \tag{4.43}$$

In all our experiments we set $\lambda_{\text{adv}} = 1$, $\lambda_c = 10$ and $\lambda_{\text{EQ}} = 0.2$

4.6.3 Experiments

(i) **Splitting the latent space**
We have observed different effects of the latent space. One of the significant issues when concatenating the two different vectors of latent space is that the neural network only uses one latent space and focuses on the interpretability of only one vector. We have observed that this is the natural behavior of the neural network. The reason behind this behavior is that the same encoder does not constraint the $\mathbf{z}_{xx}$, while for the ground motion pass through $\mathbf{z}_{yy}$ is essential for its reconstruction and variability. In previous investigations, we have try reduce to zero the specific part of the physics-based simulations :

$$||F^{xx}_{xy}(\mathbf{x})||^2_2 = ||\mathbf{z}_{xx}||^2_2 \tag{4.44}$$

The nullification of the specific part, $\mathbf{z}_{xx}$, also affects the behavior of the specific part of the broadband, $\mathbf{z}_{yy}$. Then, a drastic diminishing of the latent dimension to force the rearrangement of the latent space was not efficient either.

(ii) **Injection of the specific latent distribution**
The AdaIN equation, initially conceived for style transfer, has proven practical for accommodating one-to-many output scenarios. This strategic integration “aligns the mean and variance of the content features with those of the style features.". Particularly, the specific component of the latent space, intended for capturing high-frequency information, is propagated throughout all subsequent layers during the up-sampling process via Adaptive Instance Normalization (AdaIN).

$$\text{AdaIN}(\mathbf{c}, \mathbf{s}) = \sigma(\mathbf{s}) \left(\frac{\mathbf{c} - \mu(\mathbf{c})}{\sigma(\mathbf{c})} \right) + \mu(\mathbf{s}) \tag{4.45}$$

The formula in which $\mathbf{c}$ is the content, s the style input $\sigma(\mathbf{s})$ and $\mu(\mathbf{s})$ are respectively its standard deviation and mean. We have tried to inject $\mathbf{z}_{yy}$ as a style when keeping $\mathbf{z}_{xy}$ and $\mathbf{z}_{yx}$ as common part. However, if this strategy is plausible for managing one-to-many mapping, this performance significantly affects the reconstructions. This tentative approach has not been adapted to our objectives.

(iii) **Variability of the representations**
To improve reconstruction quality and output variability, we investigated decoder configurations. We tested the classic Conformer decoder and variants with AdaIN and U-Net (Conformer+AdaIN, Conformer+AdaIN+UNet). These variations aimed to observe how specific aspects are passed to the generator (G_y) and interpreted.

(iv) **Ablation** We have removed the explicit cycle in the hybrid loss (View Equation 4.46).

$$\mathcal{L}_{\text{hybrid}}^{yx} = \text{FFL}(\mathbf{y}, G_y(F_{xy}^{xx}(\mathbf{x}), F_{xy}^{yy}(\mathbf{y}))) + \text{HSL}(\mathbf{y}, G_y(F_{xy}^{xx}(\mathbf{x}), F_{xy}^{yy}(\mathbf{y}))) \tag{4.46}$$

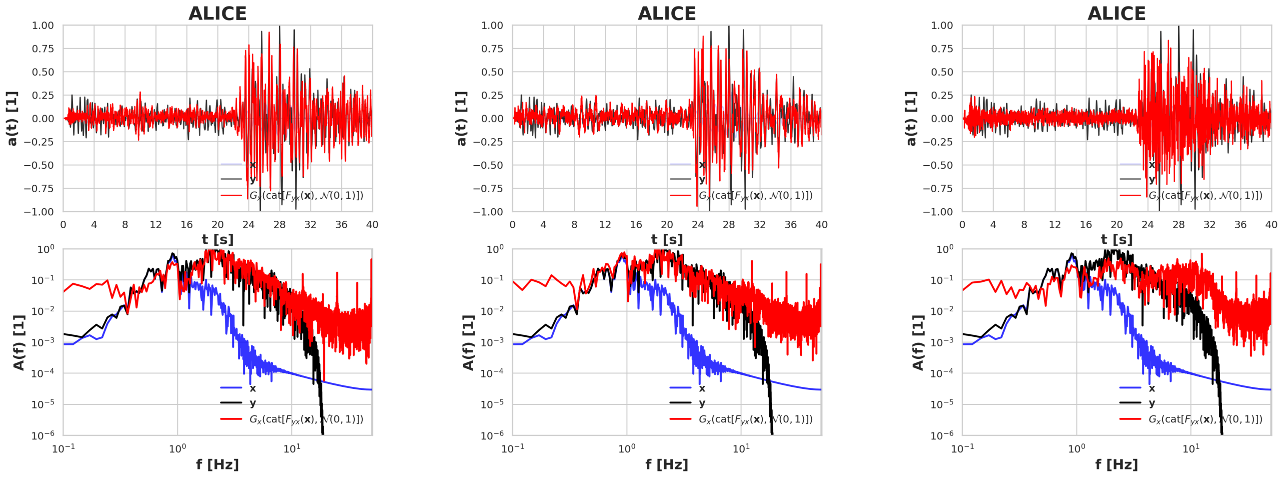
This ablation aims to test the validity of this loss for training. However, our tests reveal that the performance of the hybrid mapping is not affected, expressing the robustness of our architecture.

4.6.4 Results

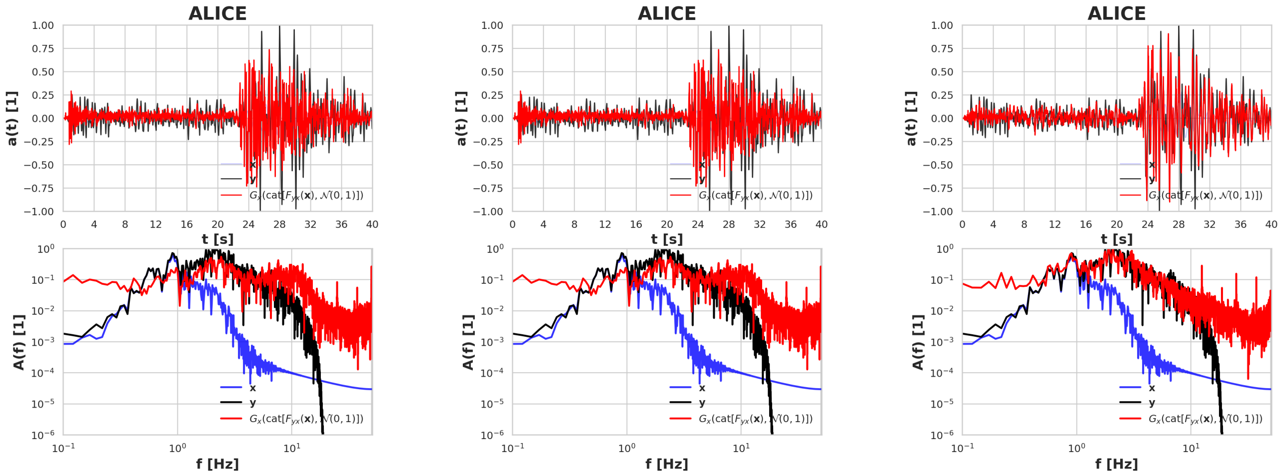
(i) **Comparison with Previous method**
This novel architecture surpasses this limitation of the previous strategy constraint in the one-to-one mapping. Such architecture could map one-to-many, as illustrated in Figure 4.29.

Moreover, its architecture validates our objectives' specific criteria (reconstructions, Gaussian Generations, in the Section § 4.6.1). Objectives I, II, IV, V, VI, and VII are considered satisfied, as the previous solution. The reader will accept that these results are verified because this novel architecture is built on top of the strategy presented in Section § 4.5.2. The comprehensive assessment of the goodness-of-fit (G.O.F) for diverse solutions is presented in Figure 4.31, providing a detailed overview of the model's performance. In the figure, we have made a histogram to view which architecture is more adapted for hybrid mapping. The comparison is between $G_y(F_{xy}(\mathbf{x}), F_{xy}(\mathbf{y}))$ and the targeted broadband $\mathbf{y}$. All of the architecture we have experienced performs well for the phase reconstructions. Based on our experiment, the more adapted architecture is the Generator that includes Conformer, which is more adapted for the hybrid reconstruction task.
Contrary to what one might think, the architecture,Conformer + AdaIN and Conformer + AdaIN + UNet, does not offer significant performance. Our investigation is that the AdaIN could change how we pass specific components and introduce more artifacts in this interpretation UNet could introduce more bias in the reconstruction. Even though we have trained separately the UNet and the Conformer + AdaIN Generator, with two different losses, this introduces artifact and lowers the signal's amplitude. In case of interest, view Figure 4.30.

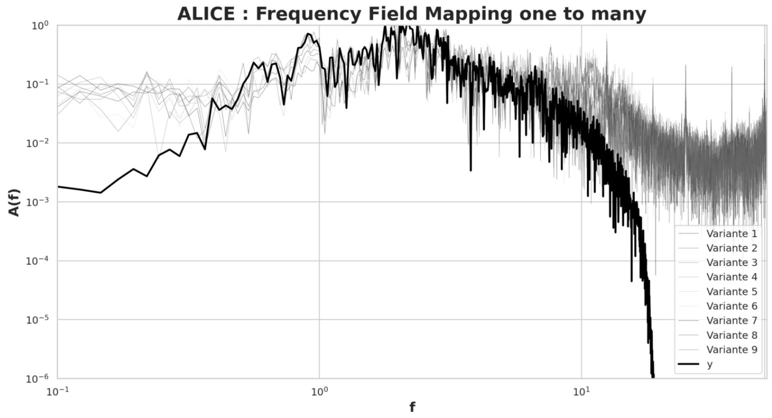
This breakthrough in mapping flexibility and validation of crucial criteria signifies an improvement of our architecture, showcasing its effectiveness in handling complex data representations and generations. View the performance of each architecture in the Table 4.6.



(a) Variant 1 (b) Variant 2 (c) Variant 3

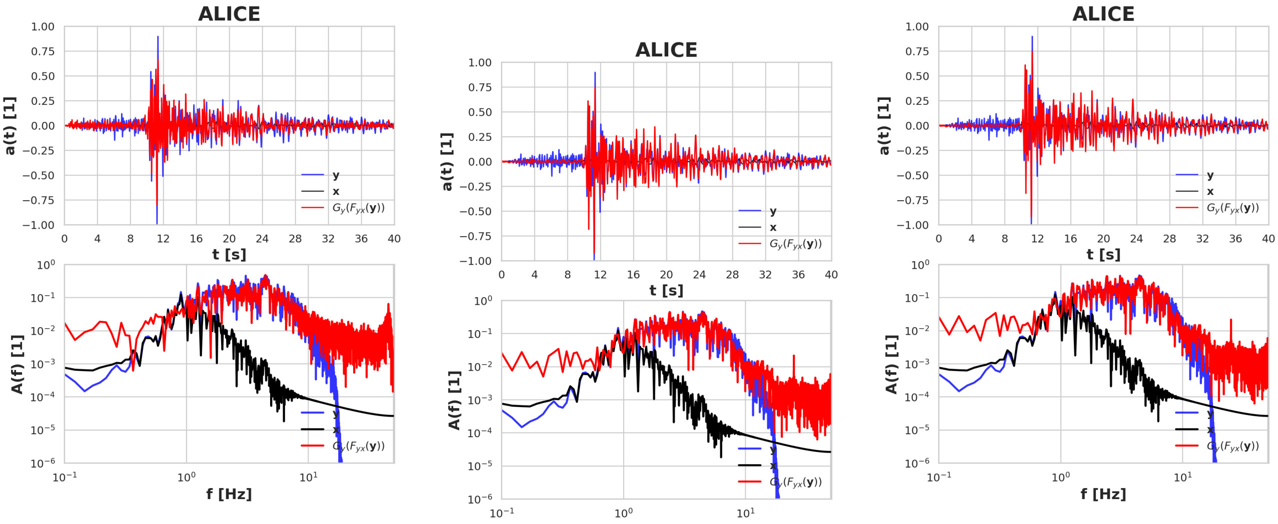


(d) Variant 4 (e) Variant 5 (f) Variant 6



(g) Frequency representation for many variants generated from the same $\mathbf{x}$. Architecture : Conformer

Figure 4.29: Here is an illustration of the multi-modal generation for the seismic data. The input $\mathbf{x}$, the high-frequency encoding aspect is replaced by a Gaussian distribution, $\mathcal{N}(0, I)$. The generated output should produce an infinity value likelihood to the targeted broadband data. $G_y(\text{cat}(F_{xy}(\mathbf{x}), \mathcal{N}(0, I)) \sim \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_\infty\}$.



(a) Quality reconstruction of a signal with the Conformer architecture

(b) Quality reconstruction of a signal with Conformer+Unet

(c) Quality reconstruction of a signal with Conformer+AdaIN+UNet

Figure 4.30: Variant issues in the reconstruction depending on the model. The architecture added fail to reconstructs, $G_y(F_{xy}(\mathbf{y}))$.

(ii) **Comparison of diverse generator architectures tested**

Exploration has been made in the research of the best-adapted neural network. To evaluate the performance of each architecture on physics-based simulations and Broadband subsets to observe which one has a better performance. In the Figure 4.32, we show which architecture performs well. As we can see, even the architecture that includes Conformer+AdaIN+Unet does not perform well enough for hybrid generation; otherwise, the quality is significantly better for the Reconstruction of physics-based simulations and Broadband signal.

(iii) **Interpretability of the latent space**

In the preceding sections, the assumption that the low-frequency information encapsulated in the common latent space should remain consistent, irrespective of whether it originates from physics-based simulations or ground motion data, has been thoroughly validated. This validation stems from an exploration conducted using PCA and t-SNE with three principal components, revealing a unified clustering pattern. This clustering pattern differs from our previous architecture with a shared latent space. Figure 4.33 presents the results of this analysis using dimensionality reduction methods.

Additionally, the visual representation in Figure 4.33 demonstrates a single clustering structure. We observe that the projection of any point of two data types is nearby. The points are from $F_{xy}^{xy}(\mathbf{x})$ (physics-based simulations) and from ground motion data ($F_{xy}^{xy}(\mathbf{y})$). Consequently, our initial assumption is unequivocally satisfied: our architecture is adept at maximizing dependencies between $\mathbf{z}_{xy}$ and $\mathbf{z}_{yx}$.

In the previous observation, some information was not used when we concatenated the data from the two latent spaces. This difficulty no longer exists. The network learns to use both parts of the space, which makes all the information produced by the network interpretable by the generator. We have, therefore, solved this problem. Overall, the new strategy employed improves the interpretability and functionality of the model and

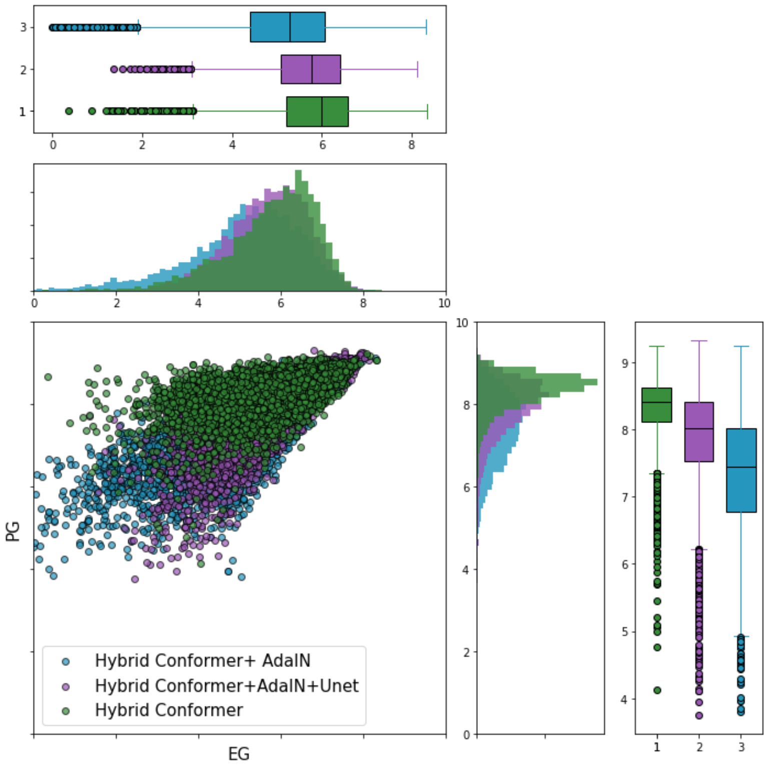


Figure 4.31: List of G.O.F for different models trained for ALICE splited latent space

Architecture	#Param. (M)	Broad.		PBS		Hybrid	
		EG	PG	EG	PG	EG	PG
Conformer	219,44	5.70 ± 1.07	8.21 ± 0.46	6.20 ± 1.35	9.13 ± 0.23	5.80 ± 1.07	8.32 ± 0.46
Conformer+AdaIN	217,40	6.41± 0.94	8.47 ± 0.43	5.52 ± 1.37	9.22 ± 0.23	5.12 ± 1.34	7.36 ± 0.87
Conformer+AdaIN+Unet	231,00	6.62 ± 0.80	8.72 ± 0.40	6.11 ± 1.39	9.21± 0.19	5.69 ± 0.97	7.87 ± 0.75

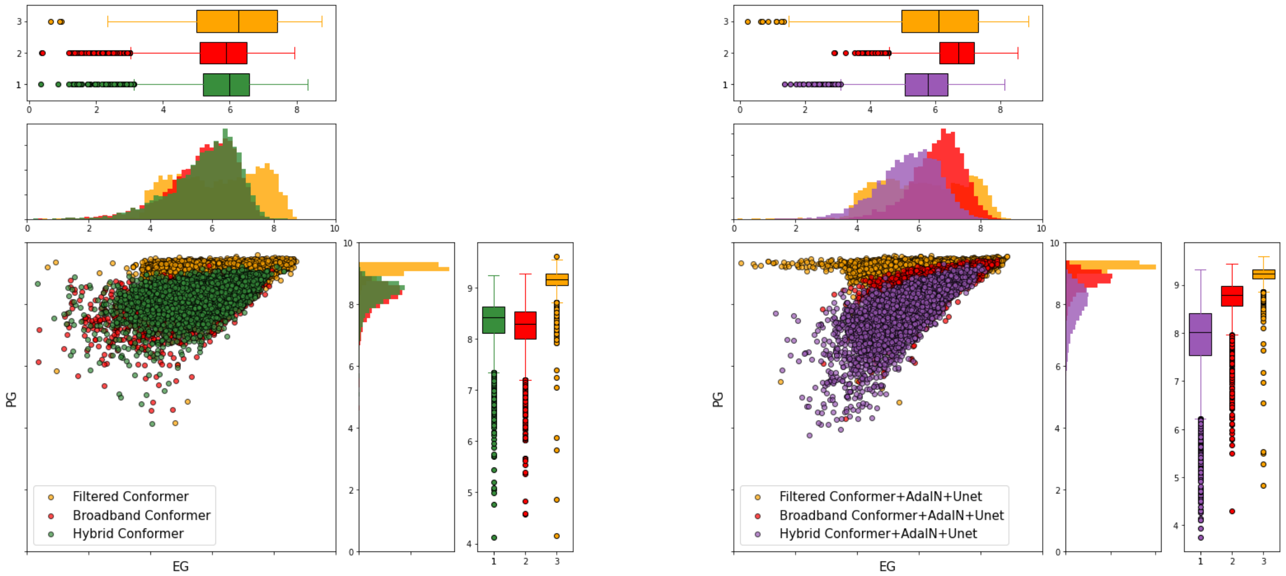
Table 4.6: Table of Architecture of Generators for strategy Unified training ALICE with splited latent space

proves its robustness in hybrid reconstruction. We consider objective VII from Section §4.6.1 to be satisfied.

4.7 Conclusion and Perspective

In conclusion, our investigation into data representation and generation methodology has been conducted with several noteworthy advanced techniques to understand and circumvent different bottlenecks and limitations. Initially, the SeismoALICE was designed to map any physic based on broadband data and reversely any recorded acceleration to simulation outputted acceleration. Face the limitations of this previous strategy, notably the duplicate resources and the contradictory loss, we have investigated and built a novel and more suitable architecture for times series, capable of viewing long-range-dependencies to address the challenge of reconstruction and generation of ground motion acceleration and also for physically based data. A better understanding of each part of the problem enables and provides the keys to addressing the hybrid mapping so that each abstract representation caught in this latent value is mathematically interpretable, particularly in achieving a one-to-many mapping for plausible outputs, crucial for civil engineering applications.

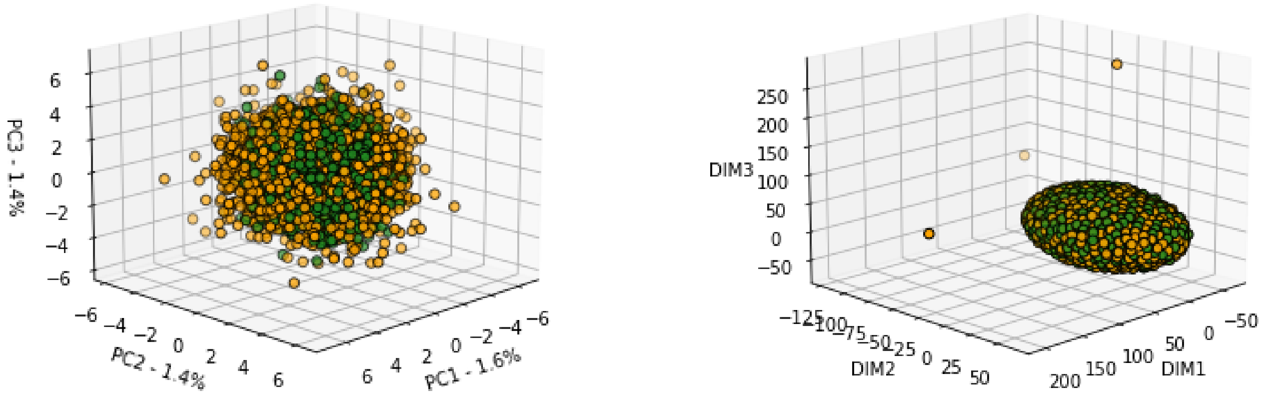
Our first investigation to overcome previous limitations was centered on developing a sur-



(a) List of G.O.F for the architecture Conformer

(b) List of G.O.F for With the architecture Conformer+AdaIN+UNet

Figure 4.32: Comparison between different architectures to evaluate the performance of the neural network to predict the output values



(a) PCA results

(b) t_SNE results

Figure 4.33: We present PCA and t-SNE for the common part, that verify that $\mathbf{z}_{xy} = F_{xy}^{xy}(\mathbf{x})$ (in green) is indifferentiable from $\mathbf{z}_{yx} = F_{xy}^{xy}(\mathbf{y})$ (in yellow)

rogate strategy around a shared latent space, providing valuable insight into the performances. Based on the advancement made for our architecture design of the two groups of subsets, we have designed a unified strategy. We can satisfy the limit above mentioned, except for the one-to-many mapping.

To achieve the latter objective, we reinterpret the input data and investigate a factorized latent space to address this limitation. A common part has the task of encoding the information in the low-frequency portion of the ground motion (0-1 Hz), while the other part encodes the remaining frequency portion (1-30 Hz). Factorization of common and specific components enhances the model's interpretability and adaptability, contributing to improved performance. In addition, removing hybrid loss in the learning process has streamlined the approach, focusing on the most important components and eliminating contradictory or redundant corrections. Meanwhile, introducing the Gaussian-specific part has added variability to the results while maintaining consistency, thus enriching the generative capabilities of the model. Despite the challenges encountered in the one-to-many mapping and some performance issues, the iterative development and adjustments have led to a more sophisticated architecture better aligned with the requirements of earthquake engineering tasks and probabilistic seismic hazard assessment.

Chapter 5

Multi-Modal Signal Translation

> *"Take the first step in faith. You don't have to see the whole staircase. Just take the first step."*
>
> — William Fletcher Durant, The Prayer Power of Positive Thinking, 1959

5.1 General idea of BiCycleGAN

Let us start this chapter by recalling the major disadvantages of the generative strategies presented in Chapters 3, 4. Although the first attempted signal translation technique, namely Pix2Pix, has proved its feasibility, it proved itself not to be robust enough. Pix2Pix is sensitive to noise, and offers no understanding of the information encoded in the bottleneck. In addition, no conditional generation is possible, and this lack of richness in the data representation gives us a clear idea of its limitations.

ALICE's learning architecture has proved effective in generating hybrid signals conditioned by the filtered version of the original recordings and in an unconditional approach. However, to achieve our goals, we had to conduct complex studies, introduce regularisation parameters, and add many extra features, such as phase control, space control, and advanced information arrangement, to meet our objectives. The task of training with ALICE is, therefore, complex. For those reasons, we intend to test an architecture inspired by BiCycleGAN, according to Zhu *et al.*, 2018 [314]. This solution deserves a chapter because of the significant difference with precedently exposed methods. Compared to other GAN approaches, the most important enhancement made in BicycleGAN is introducing the bi-cycling consistency loss. This loss constrains the generator's output. With improved realism, the translation from a domain $\mathbf{x}$ to a domain $\mathbf{y}$ is possible. BicycleGAN reinforces the connection between the two domains by using the ability of the VAE to project data into a latent and disentangled Gaussian manifold. The BiCycleGAN can handle multi-modal mapping meanings and generate diverse output. This is useful in scenarios without one-to-one correspondence between the input and output domains. More than that, we could learn a shared latent space between two domains. Figure 5.1 shows the BiCycleGAN framework. BiCycleGAN is the result of a merger between two different architectures, namely VAE-GAN (Variational Autoregressive-Encoder and Generative Adversarial Network) and cLR-GAN (Conditional Latent Regressor) [315, 112, 314]. In the following, the

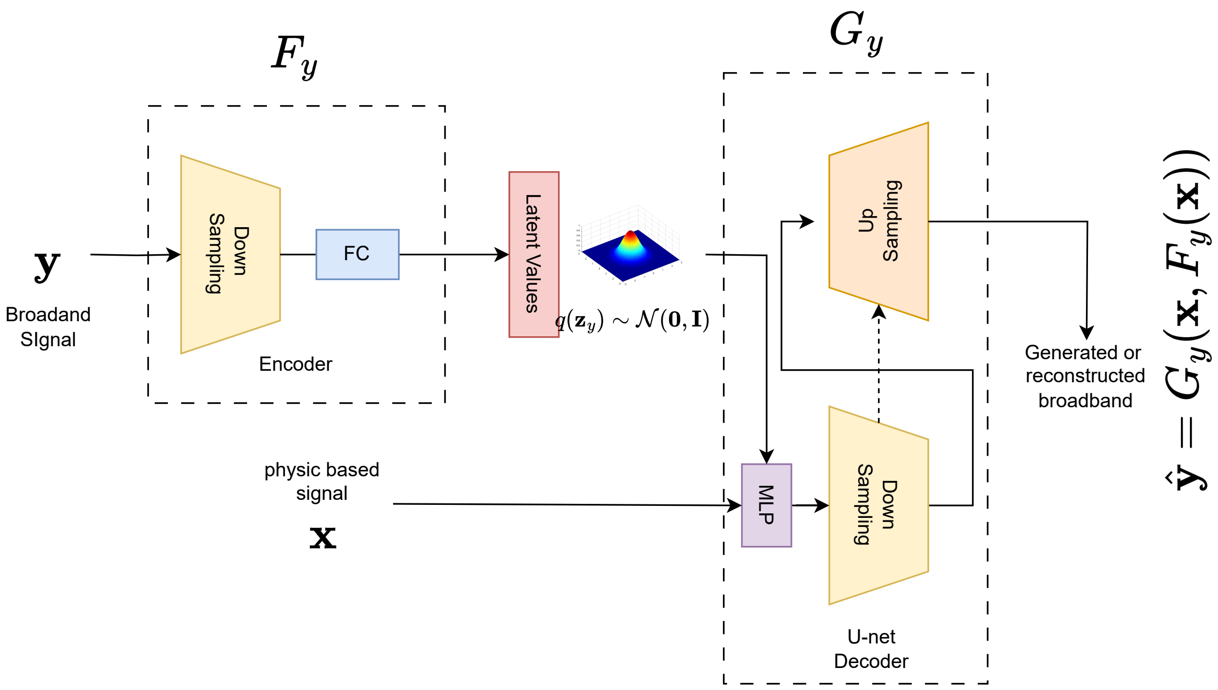


Figure 5.1: Sketch of BiCycleGAN architecture.

different aspects of BycicleGAN are described.

5.1.1 Variational AutoregressiveEncoder and Generative Adversarial Network

The VAE-GAN, also known as the Variational Autoencoder-Generative Adversarial Network, is a powerful combination of two deep learning models: Variational Autoencoder (VAE) and Generative Adversarial Network (GAN). The VAE will be used to encode information in a Gaussian space and reconstruct it later, in our case the target data is $\mathbf{y}$, which has high-frequency part. On the other hand, the GAN component simplifies the assessment of distribution pairs. It is through this data respect that multi-modal generation can happen. This intricate but critical evaluation process allows us to capture correspondences between data and generate synthetic signals that make sense, as highlighted in section 2.3.3. The VAE-GAN is trained according to the following loss function:

$$\mathcal{L}^1 = \mathcal{L}_{VAE}^{GAN}(F_y, G_y, D_{xy}) + \mathcal{L}_{KL}(F_y) + \mathcal{L}_1^{VAE}(G_y, F_y) \tag{5.1}$$

The VAE/GAN training process depends on the importance of the following term $\mathcal{L}_{VAE}^{GAN}(D_{xy}, G_y, F_y)$, where F_y is the encoder, G_y is the decoder, and D_{xy} is the joint discriminator. A VAE is here trained as a GAN, according to the following loss function:

$$\begin{aligned}\mathcal{L}_{VAE}^{GAN} = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}\left[\log(D_{xy}(\mathbf{x},\mathbf{y}))\right] + \\ \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y}), z\sim F_y(\mathbf{y})}\left[\log(1 - D_{xy}(\mathbf{x}, G_y(\mathbf{x}, F_y(\mathbf{y}))))\right]\end{aligned} \tag{5.2}$$

where G_y is represented by an U-Net. In order to minimize the distance between the encoded distribution $F(\mathbf{y}) \sim q(\mathbf{z}|\mathbf{x})$ and a Gaussian one $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, we minimize the Kullback-Leibler, D_{KL}, divergence between $q(\mathbf{z}|\mathbf{x})$ and p, that reads (We can find the definition in Equation 2.15)):

$$\mathcal{L}_{KL}(F_y) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[D_{KL}(F_y(\mathbf{y})||\mathcal{N}(0, I)))\right] \tag{5.3}$$

For our task, we prefer to manipulate an explicit loss and use the Focal Frequency Loss, suitable for audio and for any time series; see Equation 2.56.

$$\mathcal{L}_1^{VAE}(G_y, F_y) = \mathrm{FFL}(\mathbf{y}, G_y(\mathbf{x}, F_y(\mathbf{y}))) + \mathrm{HSL}(\mathbf{y}, G_y(\mathbf{x}, F_y(\mathbf{y}))) \tag{5.4}$$

5.1.2 Conditional Latent Regressor

In addition to the physic-based latent variables, the sample generation implies making the latent variable meaningful. It is then useful to constrain the generation. Instead of evaluating all potential output solutions individually, we use a Discriminator, D_y. The loss is defined as below by:

$$\mathcal{L}_{GAN}(G_y, D_y) := \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\log D_y(\mathbf{y})] + \underbrace{\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log(1 - D_y(\hat{\mathbf{y}}))]}_{\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}[\log(1-D_y(G_y((\mathbf{z}))]} \tag{5.5}$$

In this equation :

- $\mathcal{L}_{GAN}(G_y, D_y)$ is the (GAN) objective function.
- G_y is the generator to produce the broadband signals.
- D_y is the discriminator that evaluates the marginal distributions.

- $\hat{\mathbf{y}} = \mathbf{G}_{\mathbf{y}}(\mathbf{x}, \mathbf{z})$

The loss evaluate the distance between the distribution of $\mathbf{y} \sim p(\mathbf{y})$ and the distribution of $\hat{\mathbf{y}} \sim q(\mathbf{y}|\mathbf{z})$. $\hat{\mathbf{y}}$ is generated by $\mathbf{G}_{\mathbf{y}}(\mathbf{x}, \mathbf{z})$. $\mathbf{x}$ is sampled from p_G and $\mathbf{z}$ is sampled from p_Z. In the original paper of BicycleGAN [314], the cycling is forced through ℓ^1 loss, as expressed in Equation 5.6. This loss appears unimportant in our experiment enough to constrain the model and produce realistic output.

$$\mathcal{L}^{\text{Latent}} = \|\mathbf{z} - F_y(G_y(\mathbf{x}, \mathbf{z}))\|_1 \tag{5.6}$$

5.1.3 Loss function

The objective function adopted to train BicycleGAN for signal-to-signal translation can be dissected into two main components, each playing a significant role in our target multi-modal generation task. This combination includes aspects from VAE-GAN and cLR-GAN frameworks. The expression of the loss function reads:

$$\begin{aligned} V(D^*_{xy}, G^*_y, F^*_y) = {} & \arg \min_{G_y, F_y} \max_{D_{xy}} \mathcal{L}^{\text{VAE}}_{\text{GAN}}(G_y, D_{xy}, F_y) \\ & + \lambda \mathcal{L}^{\text{VAE}}_1(G_y, F_y) \\ & + \arg \min_{G_y} \max_{D_y} \mathcal{L}_{\text{GAN}}(G_y, D_y) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(F_y) \end{aligned} \tag{5.7}$$

$\mathcal{L}^{\text{VAE}}_{\text{GAN}}$ is an adversarial GAN-style loss over the joint distribution, $\mathcal{L}^{\text{VAE}}_1(G_y, F_y)$ is the reconstruction loss for hybrid signals and $\mathcal{L}_{\text{GAN}}$ the loss on the marginal distribution. $\mathcal{L}^{\text{latent}}_1$ stands for the reconstruction of the latent space, and $\mathcal{L}_{\text{KL}}$ is the Kullback-Leibler Divergence that enforces the latent space to match a standard normal distribution.

5.1.4 Architectures of BicycleGAN

Encoder

The encoder, F_y is composed of 1D-convolutional layer inspired from VGG (Very Deep Convolutional Networks) [316, 317]. We employ the same skeleton designed for recognition, Residual Network. The residual Block stabilizes the training. The data passed through several convolutional layers, multiplying the number of channels by two and dividing the number of the times step by 2. Ultimately, we make a flattened and linear layer projecting the output features to 128 hot-encoded values. View in Figure 5.2. View more details about the output in Table 5.1.

Layer	Opeation	Input Size	Fonction Activation	Stride	Output
ConvBlock I	Downsampling	[3, 4096]	LeakyReLU	2	[16, 2048]
ConvBlock II	Downsampling	[16, 2048]	LeakyReLU	2	[32, 1024]
ConvBlock III	Downsampling	[32, 1024]	LeakyReLU	2	[64, 512]
ConvBlock IV	Downsampling	[64, 512]	LeakyReLU	2	[128,256]
ConvBlock V	Downsampling	[128,256]	LeakyReLU	2	[256,128]
ConvBlock VI	Downsampling	[256,128]	LeakyReLU	2	[512,64]
Reshape	’–	[512,64]	’–	’–	[32768]
Linear Layer for mean	Projection	[32768]	’–	’–	[128]
Linear Layer for logvar	Porjection	[32768]	’–	’–	[128]

Table 5.1: Architecture of the Encoder of BicycleGAN

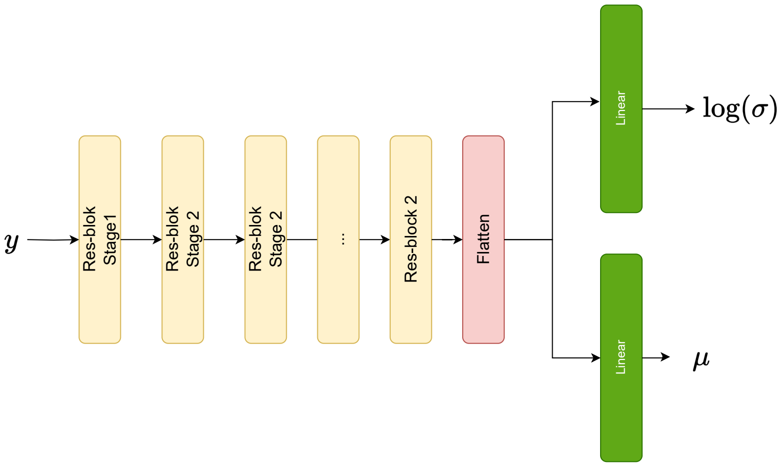


Figure 5.2: Design architecture of the Encoder, F_y

Generator

In contrast, G_y uses a U-net architecture with latent variables and physics-based simulations as inputs. However, experiments have shown that simple concatenation of the dimensions of latent variables and physics-based simulation leads to some peculiarities in the generator's behavior. To overcome this problem, another method was used: direct injection of values into the latent space.

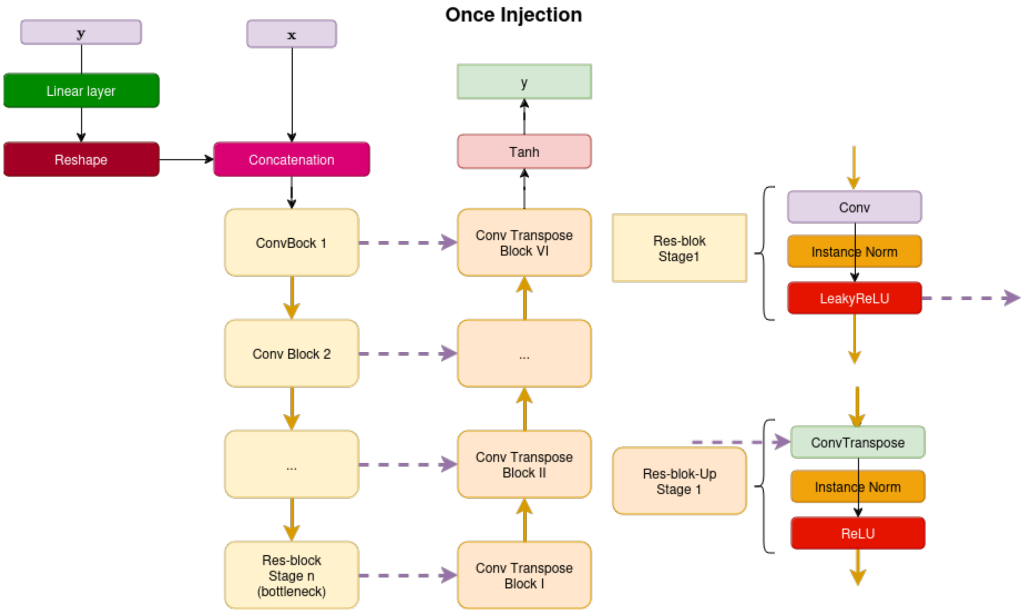
Latent space injection should be handled with utmost care due to various considerations. Up until now, two types of injections have been tried.

In the first type, known as a single injection, the latent space variable $\mathbf{z}$, a fully connected layer projects onto a 4096-dimensional space. The resulting vector is then added as an additional channel to the input signal. This injection occurs before subsampling by the convolutional layers and continues until subsampling for output prediction $\mathbf{y}$. In Figure 5.3 (a), the latent space is projected from a vector 1-dimension 128 to another dimension space the same as the length of the physic-based. This latter is after concatenated. Furthermore, the UNet architecture is used.

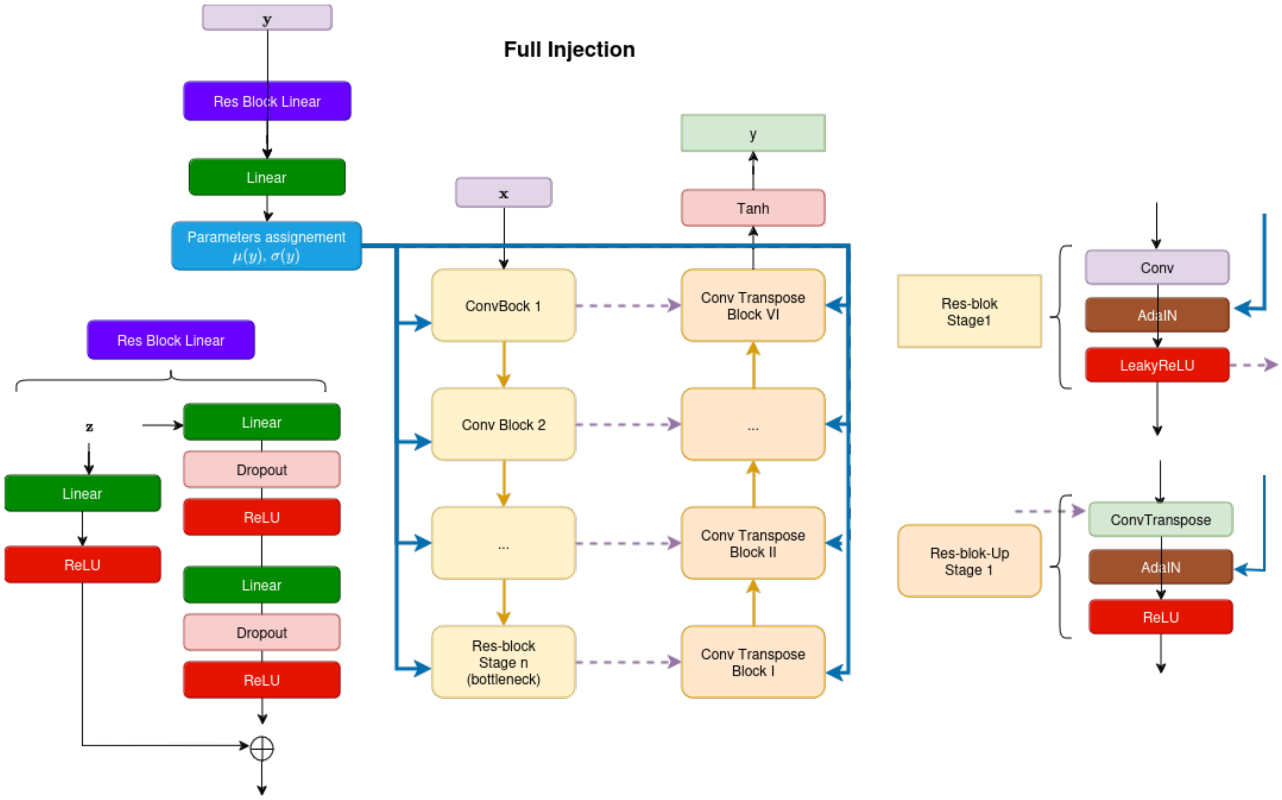
In the second type of injection, information from the latent space is projected and then re-injected into the various layers of the UNet Generator architecture. This approach involves using a technique already mentioned in the previous chapter, injecting latent space values using AdaIN (Adaptive Instance Normalisation). As we will see later, this technique is more efficient and has proved its worth. In Figure 5.3 (b), we use residual Dense Block to extract relevant feature information from the latent space. After, the corresponding output passes through a linear layer to expand the dimension of the vector. A Parameter assignment task operates different linear operations to have the mean, $\mu(\cdot)$, and standard variation, $\sigma(\cdot)$ (see Equation 4.45) the same shape as the Adaptative Instance Norm. Another difference between the once-injection and full-injection methods is the output's normalization. This latter Normalization remains the parameter that has the task of affecting the output. We have injections at all different levels of the architecture (either the downsampling part or the upsampling part).

Discriminator

We use a discriminator to evaluate the quality of the reconstruction of the signal. We use a PatchGAN-based Discriminator [298] for VAE. In the input, the two different types of data are passed. Different features are extracted, but the final values are not a score for each pair of



(a) Once injection of the latent variable, **z**



(b) Full Injections of the latent variable, **z**

Figure 5.3: Different type of Injection for the Generator, G_y

samples, which was done in previous architecture, but a grid of values. The BCEWithlogit will be used to distinguish which pair should be considered true and which is false.
For cLR, we use a classic discriminator based on DCGAN [318], which receives the architecture of the input. The information is passed through different CNN layers, reshaped, and a Linear layer that gives a score. The BCEWithLogit is used also.

5.1.5 Experiments and Results

As explained in the last section, the selected architectural framework must accomplish two key objectives. First, it must be able to reconstruct raw data expertly. Second, one-to-many mapping needs to be possible. The figure below shows how well each method accomplishes that second task. (see Figure 3) We found that full-injection was better for one-to-many mapping and did a great job of respecting the phases naturally. However, BicycleGAN has some downsides. Unfortunately, it carries some artifacts and introduces output values that don't match up with seismic signals, Figure5.5. Find a list of G.O.F for BiCycleGAN 5.6.

Architecture	#Param. (M)	Hybrid	
		EG	PG
Once injection	15,00	4.67 ± 1.05	6.11 ± 1.25
Full injection	16,07	6.87 ± 1.28	8.08 ± 1.11

Table 5.2: Architecture performance of BicycleGAN

5.2 Multimodal Unsupervised Signal Translation

5.2.1 Limitation of BiCycleGAN and SeismoALICE factorized latent space

(i) **Limitation of BicycleGAN**

Although BiGycleGAN has demonstrated its effectiveness in facilitating multi-modal translation and generation tasks while maintaining a good score for hybrid reconstruction, it is imperative to recognize certain limitations inherent in its design. Notably, the BiCycleGAN is only possible in one direction because it performs well the mapping one-to-many (from $\mathbf{x}$). However, it cannot do the reverse mapping, *i.e.* retrieves the corresponding PBS from the $\mathbf{y}$.

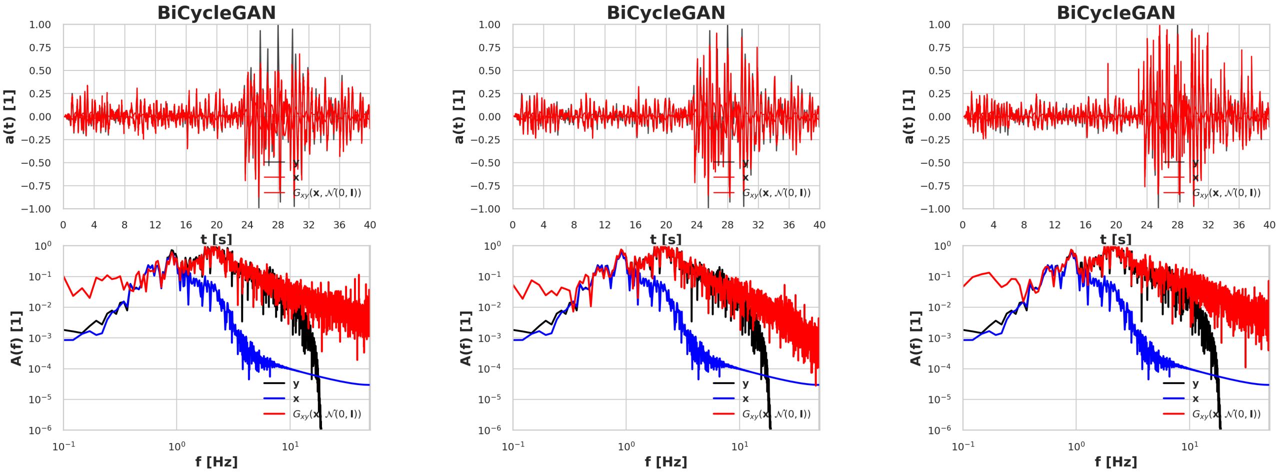
Moreover, this architecture introduces the artifact in the outputs, which could not be plausible in analyzing hybrid generations. See Figure 5.5 in BiCycleGAN. The model could suffer from handling significant variations. Handling complex large domains with this architecture could cause the model to vary only in one aspect and add noise to the other part that it does not understand.

(ii) **Limitation of Signal-to-Signal translation with ALICE**

(1) **Latent space design**

The architecture of SeismoALICE, designed to perform the mapping one to many, adopts a split representation of the latent space. In this design, low frequency is assumed to be passed through the common part ($\mathbf{z}_{xy}$ or $\mathbf{z}_{yx}$), and for any greater information, while the specific part is used for handling higher-frequency details.

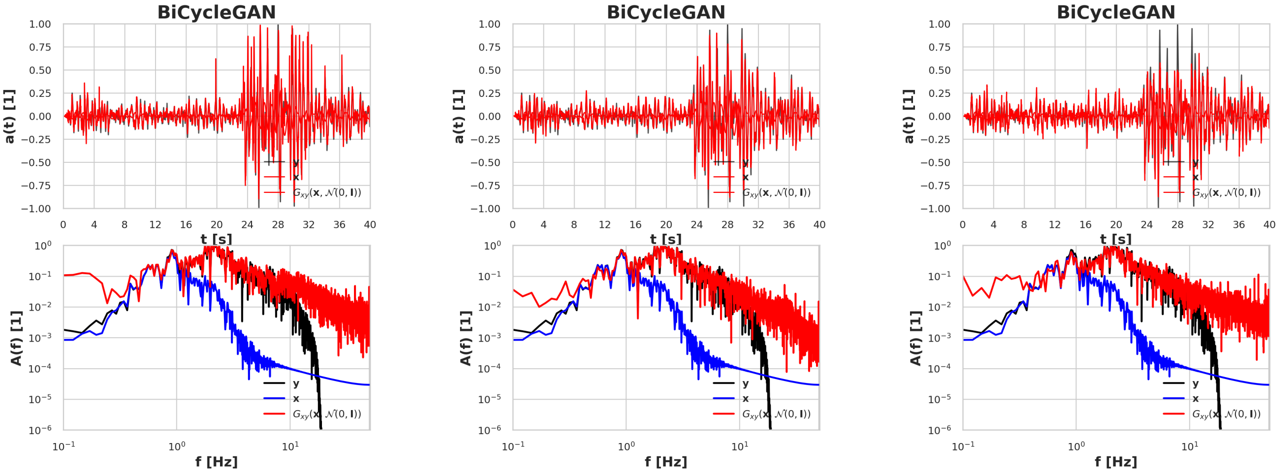
Various methods, such as VQGAN, VQ-VAE [231] and approaches such as codec [319] or residual vector quantizers (RVQ) [227], have explored discreet latent values, to circumvent the restriction of continuous Latent values to gain a better performance. In contrast, our ALICE formulation opts for continuous latent values and rearranges abstract information extracted from features. This task is carried out in a way that



(a) Variant 1, $G_y(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$

(b) Variant 2, $G_y(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$

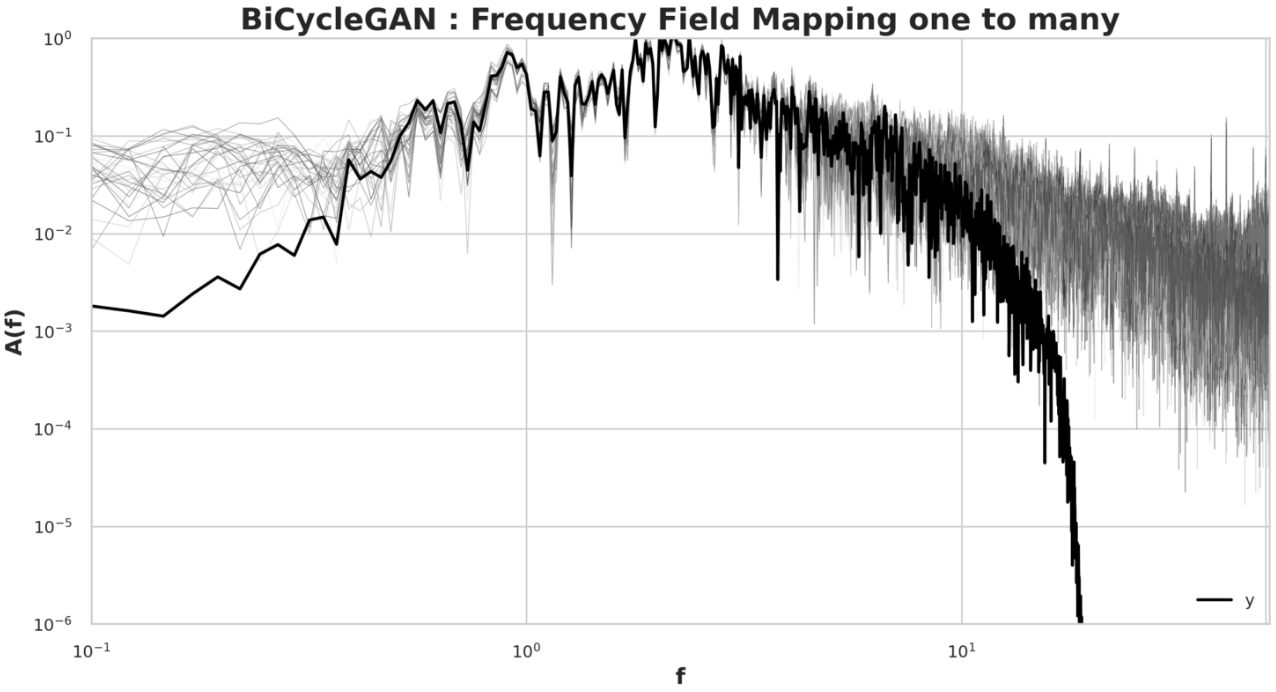
(c) Variant 1, $G_y(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$



(d) Variant 1, $G_{xy}(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$

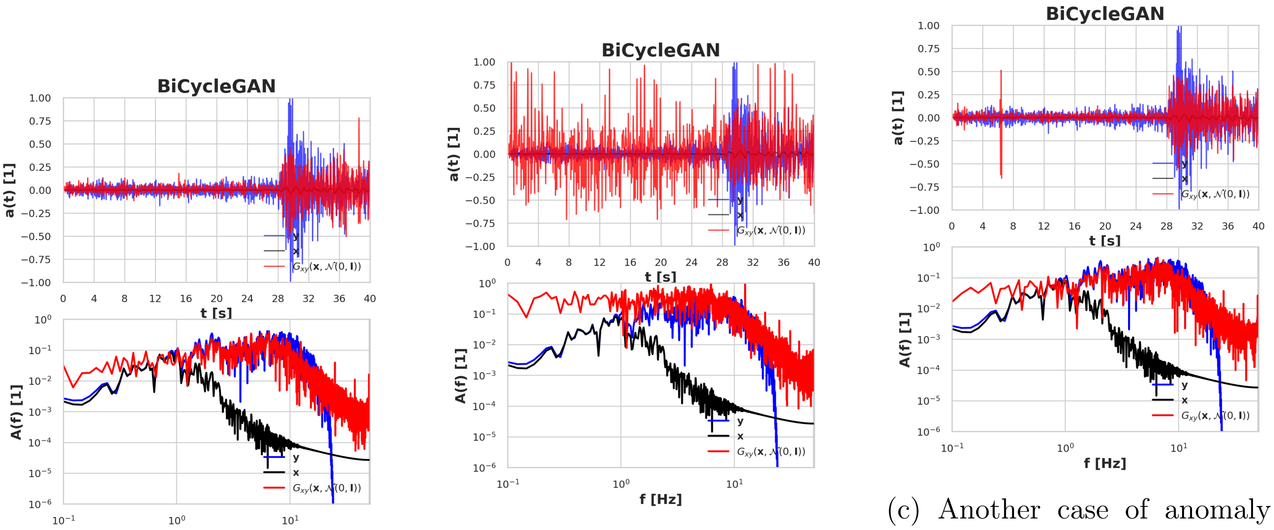
(e) Variant 1, $G_y(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$

(f) Variant 6, $G_y(\mathbf{x},\mathbf{z})$, with $z \sim \mathcal{N}(0,\mathbf{I})$



(g) View of the generated signal compared (in grays) to the targeted ground motion signal in black

Figure 5.4: Mapping one to many using BicycleGAN using full_injection technique



(a) Normal generated output

(b) Case of anomaly generated output

(c) Another case of anomaly detected in the generated output.

Figure 5.5: Case of Anomaly generated output from the BiCycleGAN strategy, illustration with once injection

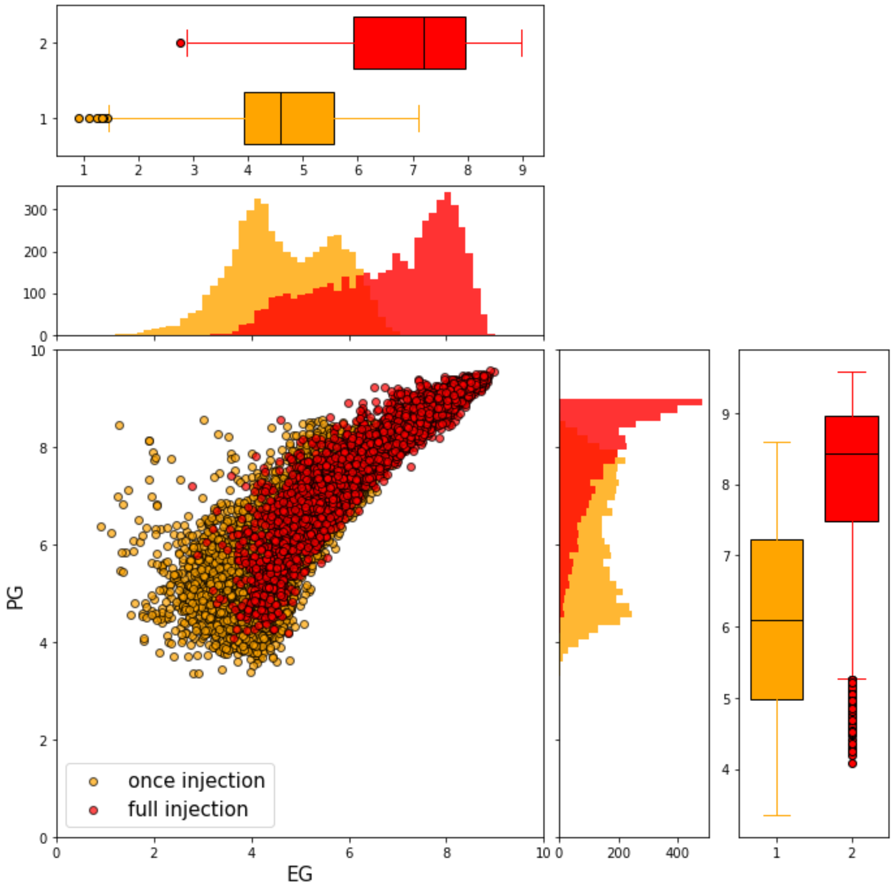


Figure 5.6: List of G.O.F for the BiCycleGAN. The test have been made on 12,800 signals 3D signals

corresponds to the specific objectives of the application. However, it is recognized that the exact nature of this rearrangement may differ from the assumptions.

(2) **Interpretability**
We have put much effort into making the latent space easier to interpret, but it is still a big black box. Moreover, we have yet to learn what the encoder is doing. We have proven in the previous section that the encoder is important for its ability to map $\mathbf{x}$ and $\mathbf{y}$ from different subsets onto the same common part $\mathbf{z}_{xy}$ and $\mathbf{z}_{yx}$ respectively. However, because the, $\mathbf{z}_{xx} = F_{xy}^{xx}(\mathbf{x})$ is not constraint, we need more interpretation.
The ambiguity could arise in several possible ways: the encoder simply rearranges information to make it clearer by moving relevant features or patterns closer together or separating them. This interpretation does not exactly fit with our intuition about "clarity," but it does seem like an improvement over noise.
Another possibility is that $\mathbf{z}_{xx}$ contains many redundant or duplicated versions of some interventions/infractions, which are not necessary for a generation but are hard to eliminate without giving up on transfer learning entirely (and just training separate models for each intervention).
Alternatively, there may be no useful information in $\mathbf{z}_{xx}$! In this case, adding more capacity to F_{xy} will allow us to improve performance when other parts of F_{xy} are under-trained. However, once those other parts get enough capacity, there will be no benefit from pushing more information into $\mathbf{z}_{xx}$.

5.2.2 Methodology

These constraints forced us to explore alternative models. The Multimodal Unsupervised Signal Translation (MUST) model can create a one-to-many mapping and produce outputs with higher fidelity to the underlying content. See Figure 4 for an overview of the MUST Encoder-Decoder architecture.

One key aspect of the MUST model is that it carefully decouples style and content-encoding at the input and output. The style is Gaussian by default, while the content should be low-frequency [0-1Hz]. We do not determine this distribution in this architecture but let the neural network learn it internally. We add a loss term minimizing the distance between content extracted from PBS and Ground Motion as in [320]. This approach can be seen as an extension of Multimodal Unsupervised Image-to-Image Translation (MUNIT). Implicit Auto Encoder [321] technique (where information is decomposed into compressible and incompressible features and adversarial loss is used to respect the same formulation as Implicit Auto Encoder), where style represents compressible features while content represents information we do not compress.

Furthermore, instead of using BicycleGAN as in previous work on MUNIT, we add low-frequency representation losses across datasets, which is a critical assumption for our later analysis. The assumption here is that broadband data contains all physical phenomena information encoded in it, so each dataset's content component contains information extracted from the physical domain even though they are different measurements physically. This assumption will be integrated into our objective functions further rigorously. For the task, one encoder, F_{xy}, and two decoders, G_x and G_y, are used. To evaluate the distributions, we use three different discriminators. D_{xy}, D_x and A series of objective functions is minimized or maximized. The equation 5.8 give :

$$\begin{aligned}\min_{F_{xy},G_x,G_y}\ \max_{D_{xy},D_{xz},D_{yz}} V =& \lambda_{\text{adv}}(\mathcal{L}_{xy}^{\text{ALI}} + \mathcal{L}_{x}^{\text{ALI}} + \mathcal{L}_{y}^{\text{ALI}}) \\ &+ \lambda_{\text{rec}}(\mathcal{L}_1^{\text{x}} + \mathcal{L}_1^{\text{y}}) \\ &+ \lambda_{\text{c}}\mathcal{L}_1^{\text{c}} + \lambda_{\text{hyb}}(\mathcal{L}_1^{\text{hyb.x}} + \mathcal{L}_1^{\text{hyb.y}})\end{aligned} \tag{5.8}$$

In this formula, the terms represent:

- $\mathcal{L}_{xy}^{\text{ALI}}$, $\mathcal{L}_{x}^{\text{ALI}}$, $\mathcal{L}_{y}^{\text{ALI}}$ are respectively the adversarial loss that evaluates the pair of distribution of for hybrid data, physic-based generation and, broadband generation
- $\mathcal{L}_1^{\text{x}}$, $\mathcal{L}_1^{\text{y}}$ are respectively the explicit cycle consistency for the physic-based data and ground-motion data.
- $\mathcal{L}_1^{\text{c}}$ is the loss that explicitly forces the content loss to be the same for the broadband and the filtered.
- $\mathcal{L}_1^{\text{hyb.x}}$ and $\mathcal{L}_1^{\text{hyb.y}}$ are, respectively, the hybrid loss to evaluate the quality of the reconstruction data.

The parameters λ_{adv}, λ_{rec}, λ_c and λ_{hyb} are the trade-off coefficients, which we have determined by experiments for best performance of the training.

5.2.3 Adversarial Loss

The adversarial loss evaluates the pair of distributions for the generated data. Rather than optimize the marginal distributions, this loss helps discriminate a pair of distributions. Meanwhile, the Encoder and the decoder are trained to produce realistic data from the input. This ability is an extension of the ALI loss introduced. Our training is particular because we do not search for the content's distribution; the Encoder will learn it by himself. The attention mechanism is only applied to the style loss, for which we assume the distribution should be classed as a Gaussian distribution. As example $\mathbf{s}_x \sim p(\mathbf{z})$. For the sack of clarity, the Encoder, F_{xy}, is split into two parts. The content part is F_{xy}^c, and the style part is F_{xy}^s. But the reader shall remember that, by definition $F_{xy}(\mathbf{x}) = (F_{xy}^c(\mathbf{x}), F_{xy}^s(\mathbf{x}))$.

The adversarial losses trained on the ground motion data help the generator G_y with the content $F_{xy}(\mathbf{y})$ and a random style $\mathbf{s}_y \sim \mathcal{N}(0, \mathbf{I})$, to output realistic ground motion distributions. So the path to retrieve the same data is as follows:

$$\mathbf{y} \xrightarrow[F_{xy}]{} \hat{\mathbf{c}}_y, \hat{\mathbf{s}}_y \xrightarrow[G_y]{} \tilde{\mathbf{y}} \tag{5.9}$$

The term, $\hat{\mathbf{c}}_y$ is an estimation of the content loss. Moreover, $\hat{\mathbf{s}}$ estimates the style loss output by the style encoder part. The target distribution is designed by $\mathbf{s}_y$. This latter is chosen randomly among the Gaussian space, which lies in a 128-dimensional space.

The loss is the flowing equation sanctions joint evaluation of the likelihood of the sample we manipulate:

$$\begin{aligned}\mathcal{L}_{\mathbf{y}}^{\text{ALI}} =& -\left(\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}\left[\log D(\mathbf{y}, F_{xy}(\mathbf{y}))\right]\right. \\ &\left.+\mathbb{E}_{\mathbf{y}\sim p_{\text{data}},\mathbf{s}_y\sim p_z}\left[\log(1 - D((G_y(F_{xy}^c(\mathbf{y}), \mathbf{s}_y), (F_{xy}^c(\mathbf{y}), \mathbf{s}_y)))\right]\right)\end{aligned} \tag{5.10}$$

Same as the raw data, the same type of adversarial loss to discriminators is for the filtered data. $\hat{\mathbf{c}}_x$ refers to the content loss. The estimation $\hat{\mathbf{s}}_x$ is used to design the outputted values from the style content. The term $\mathbf{s}_x$ is for known distribution.

$$\mathbf{x} \xrightarrow[F_{xy}]{} \hat{\mathbf{c}}_x, \hat{\mathbf{s}}_x \xrightarrow[G_x]{} \tilde{\mathbf{x}} \tag{5.11}$$

Therefore,

$$\begin{aligned} \mathcal{L}_{\mathbf{x}}^{\mathrm{ALI}} = &-\left(\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}\left[\log\sigma(D_{xz}(\mathbf{x}, F_{xy}(\mathbf{x})))\right]\right. \\ &\left.+\,\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}, \mathrm{s}_x\sim p_z}\left[\log(1-\sigma(D_{xz}((G_x(F^c_{xy}(\mathbf{x}), \mathrm{s}_x), (F^c_{xy}(\mathbf{x}), \mathrm{s}_x)))))\right]\right) \end{aligned} \tag{5.12}$$

Before getting closer to the mathematical distribution we want, each value of $\mathbf{x}$ is presumptively related to a distinct $\mathbf{y}$. These hybrid values produced should have identical distributions. We also assume that $\mathbf{y}$ and $\mathbf{x}$ target the same content loss; this has been covered in the previous sections. So the targeted values are then :

$$\begin{aligned} \mathcal{L}_{\mathbf{xy}}^{\mathrm{ALI}} = &-\Big(\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\mathrm{data}}, s_y\sim p_z}\left[\log\sigma(D_{xy}(\mathbf{x}, G_y(F^c_{xy}(\mathbf{y})), \mathbf{s}_y))\right] \\ &+\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{\mathrm{data}}, \mathbf{s}_x\sim p_z}\left[\log(1-\sigma(D_{xy}(G_x(F^c_{xy}(\mathbf{x}), \mathbf{s}_x), \mathbf{y})))\right]\Big) \end{aligned} \tag{5.13}$$

5.2.4 Content Loss

For the content loss, we use the Focal Frequency Loss, on the content loss. The content information is a 3D tensor with batch, channel, and length. The content Encoder does downsampling, which is improved by using the conformer for the tasks.

$$\mathcal{L}^c = \mathrm{FFL}(F^c_{xy}(\mathbf{x}), F^c_{xy}(\mathbf{y})) + \mathrm{HSL}(F^c_{xy}(\mathbf{x}), F^c_{xy}(\mathbf{y})) \tag{5.14}$$

5.2.5 Explicit cycle consistency

We perform explicit cycling for PBS

$$\mathcal{L}_1^{\mathrm{x}} = \mathrm{FFL}(\mathbf{x}, G_x(F_{xy}(\mathbf{x}))) + \mathrm{HSL}(\mathbf{x}, G_x(F_{xy}(\mathbf{x}))) \tag{5.15}$$

We perform explicit cycling for the raw data

$$\mathcal{L}_1^{\mathrm{y}} = \mathrm{FFL}(\mathbf{y}, G_y(F_{xy}(\mathbf{y}))) + \mathrm{HSL}(\mathbf{y}, G_y(F_{xy}(\mathbf{y}))) \tag{5.16}$$

We perform explicit cycling for hybrid

$$\begin{aligned} \mathcal{L}^{\mathrm{hyb.x}} = &\,\mathrm{FFL}(G_x(F^c_{xy}(\mathbf{y}), F^s_{xy}(\mathbf{x}))) \\ &+\mathrm{HSL}(G_x(F^c_{xy}(\mathbf{y}), F^s_{xy}(\mathbf{x}))) \end{aligned} \tag{5.17}$$

the broadband hybrid

$$\begin{aligned} \mathcal{L}^{\mathrm{hyb.y}} = &\,\mathrm{FFL}(G_y(F^c_{xy}(G_x(F_{xy}(\mathbf{x}))), F^s_{xy}(\mathbf{y})), \mathbf{y}) \\ &+\mathrm{HSL}(G_y(F^c_{xy}(G_x(F_{xy}(\mathbf{x}))), F^s_{xy}(\mathbf{y})), \mathbf{y}) \end{aligned} \tag{5.18}$$

5.2.6 Architecture

Generators

(i) **Encoder Architecture**
The Encoder Architecture in our model features different levels of Residual Blocks to facilitate the down-sampling. It resembles the Encoder of SeismoALICE, except for a critical distinction: it incorporates two separate branches. One of these branches is designated as the Style Encoder, while The other is called Content Encoder. Importantly, these branches are not connected; the input goes in the two pipelines simultaneously (view Figure 5.7). In the Content part, only Conformer blocks are included, emphasizing a specific type of architecture for this portion of the encoding process. The aforementioned Conformer signifies a deliberate choice to capture and process content-related information.

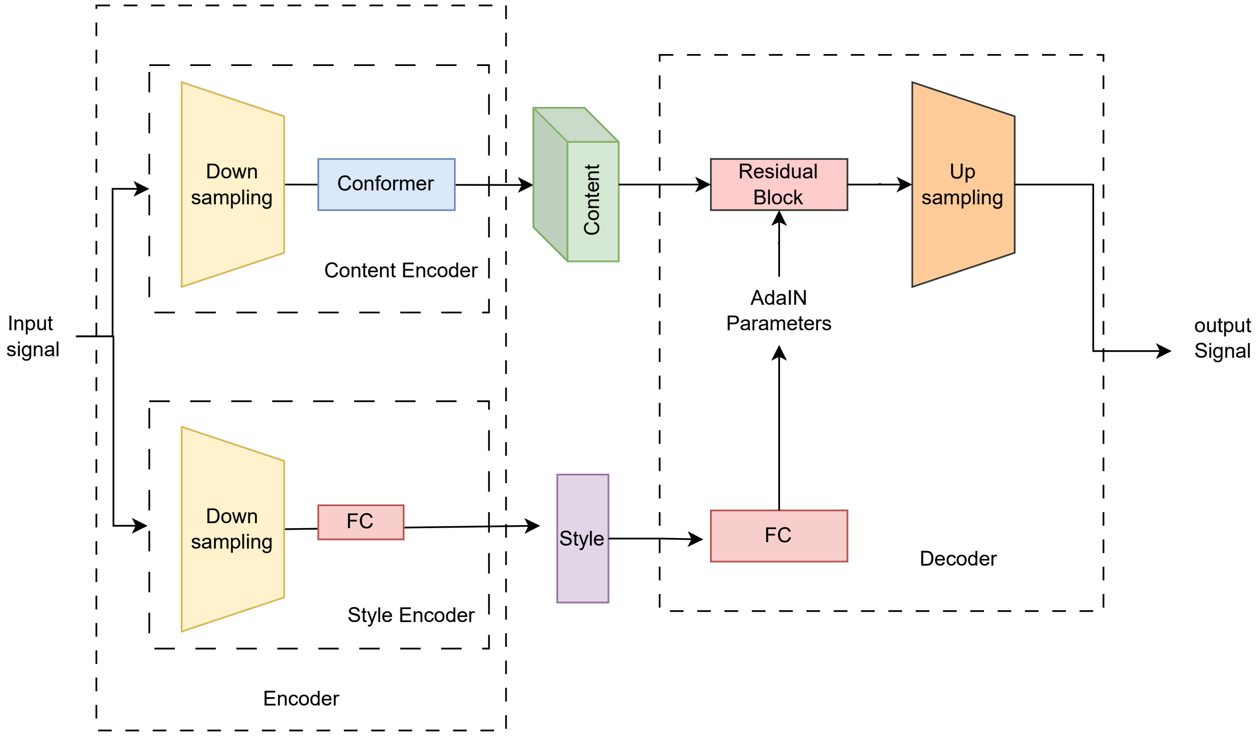


Figure 5.7: We present the models for the multi-modal unsupervised signal translation. We show that the encoder decomposed the information into two parts: the content and the style.

This dual-branch design allows the model to simultaneously extract and encode both style and content information independently, avoiding affecting each weight's part or introducing irrelevant processes in the architecture; such behavior is not easy to detect and often leads to ill-designed architecture. Also, the separation can give the model a more nuanced understanding of the input data, enhancing its capacity to disentangle and manipulate different aspects during the training process.

Therefore, the choice of this architecture reflects a careful consideration of how style and content information (respectively partly compressed and partly uncompressed) contribute to the overall representation of the input data and how processing them separately can limit the model's ability to learn effectively.

(ii) **Decoders Architectures**
The Decoder Architecture receives both style and content. On the one hand, the content

is directly passed through the convolution layer, which performs the up-sampling process. On the other hand, the style in counterpart is not directly passed; instead, it is conveyed through a series of Residual Dense blocks to extend its dimension before the application Adaptive Instance Norm (AdaIN). See the Equation 4.45, in Chapter 4.

This technique is borrowed from StyleGAN [285],[182] [184]. The utilization of AdaIN allows style information to be injected into the content, enriching the expressiveness and diversity of the output generated. By manipulating style at a higher level in the decoder, the model provides greater control over the appearance and characteristics of the generated signals, reducing the need for more regularisation tools. The decoder is inherently designed to significantly improve the variability of our hybrid generation while preserving the quality of our reconstructions.

Discriminators

The implemented design architecture comprises three discriminators, denoted as D_{xy}, D_{xz}, and D_{yz}. This architectural framework leverages the advantageous strategy previously employed in our SeismoALICE architecture, which features a shared and factorized latent space. This strategy ensures the fulfillment of the Nash Equilibrium, with shared weights among the discriminators. In the case of MUST's implementation, slight different.

1. **joint discriminator for PBS**
   In the Figure 5.8, the Discriminator, D_{xz} as three inputs : the physic-based $\mathbf{x}$, the content $\mathbf{c}_x$ and the style $\mathbf{s}_x$. To extract information, we have exploited the same architecture as ALICE. The Discriminator will include two branches : D_{s_x} and $D_{s_{z_x}}$. The first branch extracts features from $\mathbf{x}$ and will have the same architecture as the content part of the encoder F_{xy}. The second branch, $D_{s_{z_x}}$ should take account of the pair $(\mathbf{c}_x, \mathbf{s}_x)$: $\mathbf{c_x}$ is flattened before transformed to a 1-dimensional vector and the output is concatenated with $\mathbf{s}_x$, then the result passes through a residual linear block. Finally those different values (from D_{s_x} and D_{c_x}) will be concatenated. Then, we proceed to a final Residual Linear block. The output of a discriminator is the score, which will serve as the entry of the BCEWithLogit loss. Here the Equation 5.12.

$$D_{xz}(\mathbf{x}, \mathbf{c}_x, \mathbf{s}_x) = D_{xz}(D_{s_x}(\mathbf{x}), D_{s_{z_x}}(\mathbf{c}_x, \mathbf{s}_x)) \tag{5.19}$$

2. **Joint discriminator for Broadband**
   The Discriminator D_{yz} respect the same architecture. We use D_{s_y} to manage the broadband. To extract the features of the content $\mathbf{c}_y$ and the style $\mathbf{s}_y$, we use the Discriminator, $D_{s_{z_x}}$. Therefore, we have:

$$D_{yz}(\mathbf{y}, \mathbf{c}_y, \mathbf{s}_y) = D_{yz}(D_{s_y}(\mathbf{x}), D_{s_{z_y}}(\mathbf{c}_y, \mathbf{s}_y)) \tag{5.20}$$

3. **Joint discriminator for Hybrid**
   Finally, for the Discriminator dedicated to hybrid, D_{xy}, we reuse the ensemble of weights employed to extract features from the PBS and the features designed to extract features from broadband data (D_{s_y} and D_{s_x}). The results are concatenated before passing to a last residual linear block $D_{s_{xy}}$, scoring the quality of the learned distribution by the Encoder F_{xy} and the Decoder G_y and G_x. Figure 4.19 illustrates the same structure Chapter 4. The equation expressing this is formulated by:

$$D_{xy}(\mathbf{x}, \mathbf{y}) = D_{s_{xy}}(D_{s_y}(\mathbf{x}), D_{s_y}(\mathbf{y})) \tag{5.21}$$

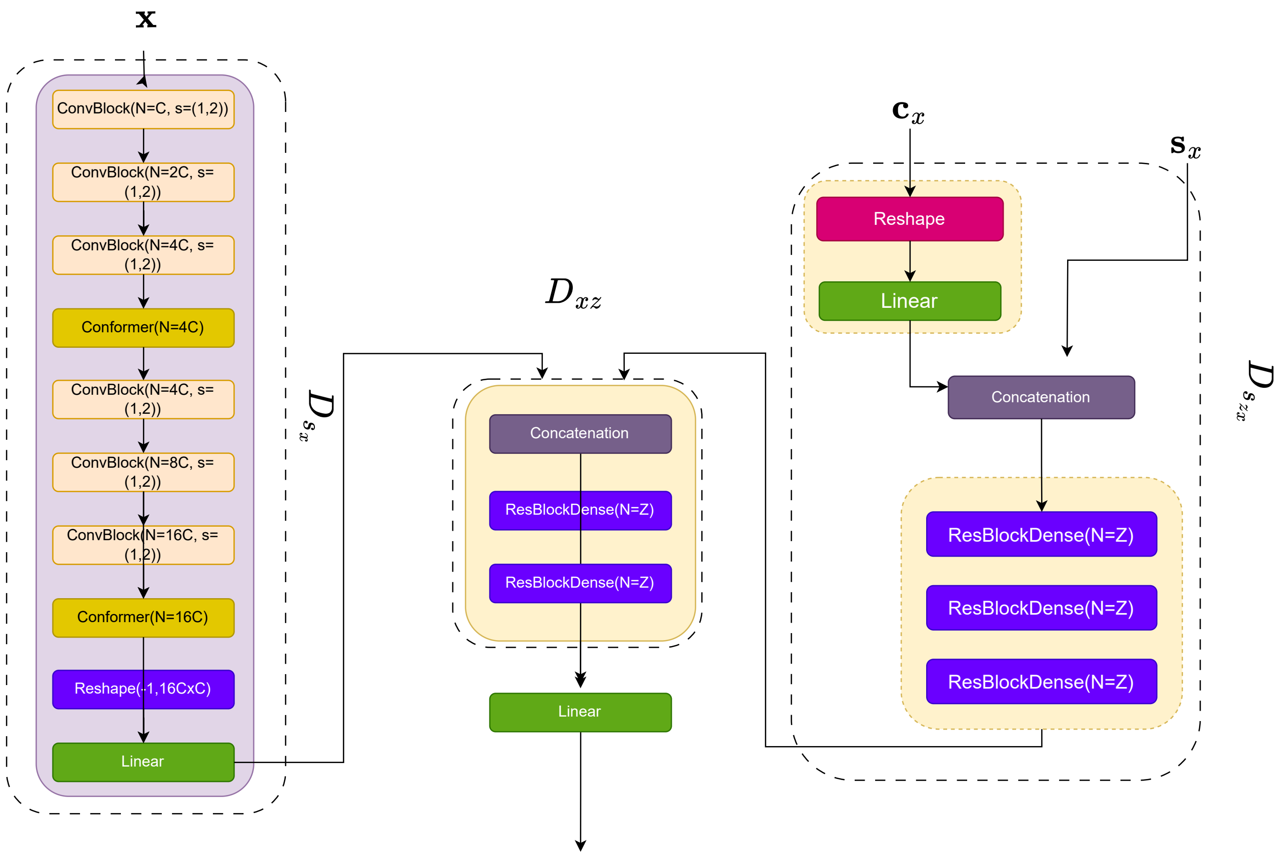


Figure 5.8: In this case, the architecture of a joint discriminator is the $\mathbf{z}_x$. The special feature of this architecture is that it considers the content, which is a three-dimensional tensor. The latter is reshaped and then concatenated with the style before moving on to the succession of residual linear blocks. The detailed architecture is similar to the one presented in chapter 4

Spectral normalization is used in all the layers of the discriminators. The optimization is proceeded via a BCEWithLogit Loss, more stable numerically than BCE.

5.2.7 Trial details

1. **Test with Constraint on style**
   Adding an adversarial loss to force the style to follow a Gaussian distribution is a logical technique for achieving the desired disentangled representation. However, our experiments found that this explicit constraint could affect the quality of the hybrid signal. We observe a degradation of the hybrid generation when we constrain the style with explicit cycling or a cross-entropy loss.

2. **Ablation of Phase picking**
   In contrast to the previous technique, SeismoALICE, the need for a phase loss is no longer necessary to guarantee the plausibility of the reconstruction. The various variants of the target signal maintain the alignment in the timing of P-Wave and S-Wave. This observation suggests the model can inherently capture and preserve the temporal relationships without the explicit need for phase loss. Interestingly, when a loss on phase is introduced, it does not yield significant results in hybrid generation. This showcases the robustness

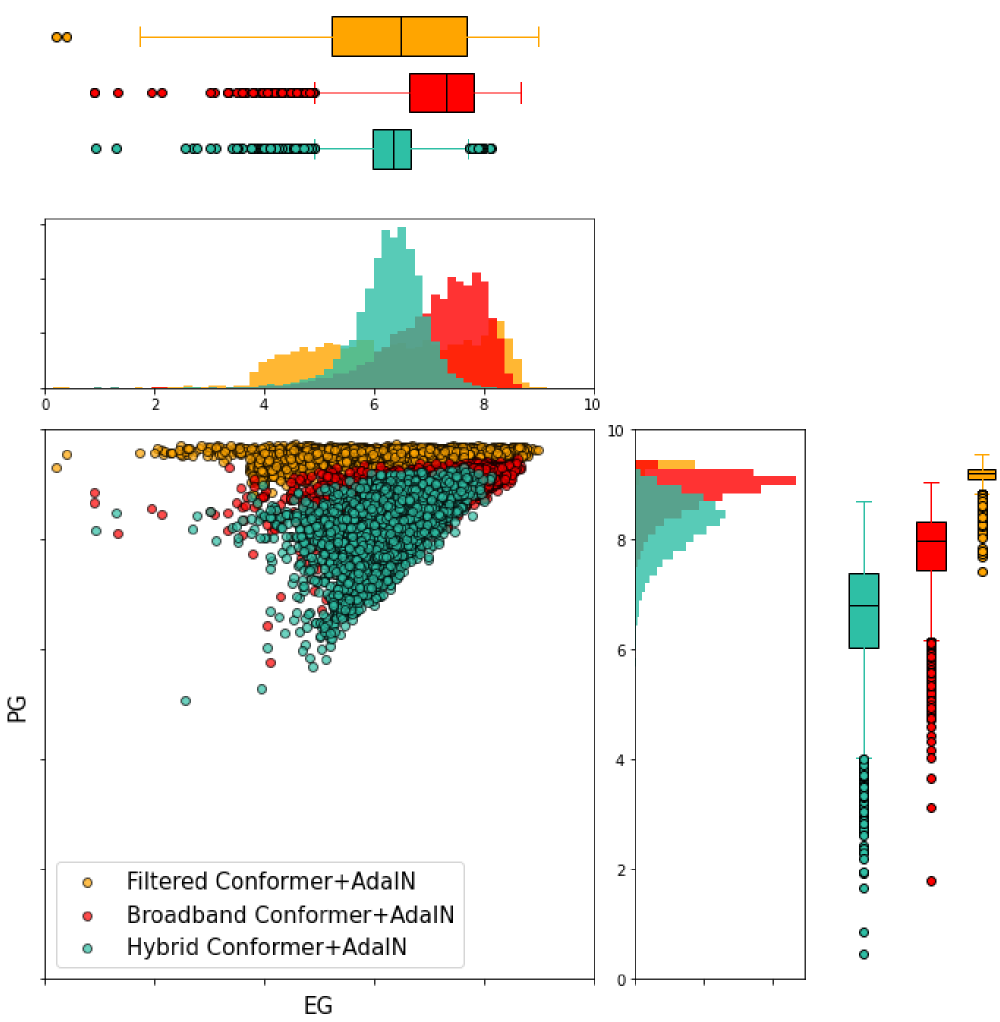


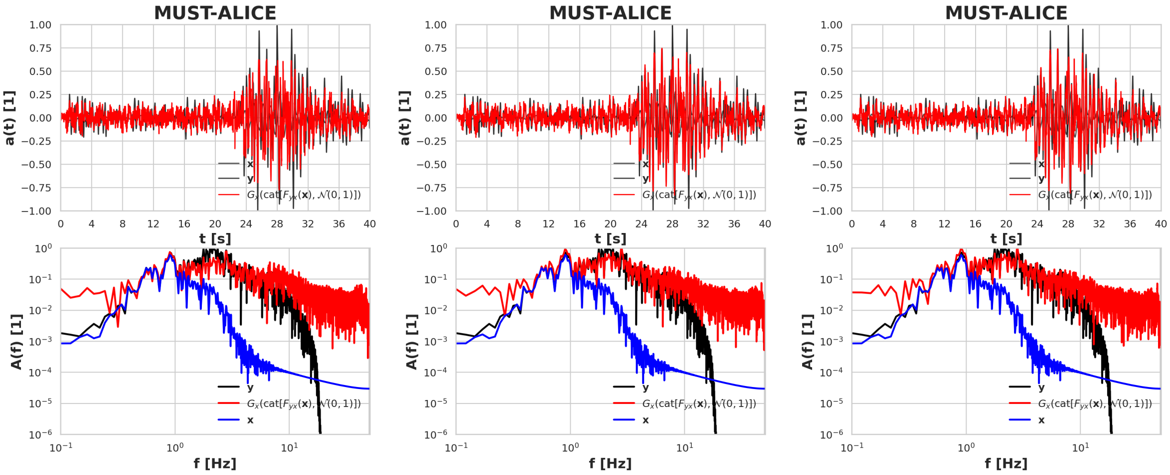
Figure 5.9: List of G.O.F for the training of MUST

of the model in the pipeline training, allowing this latter to produce complex temporal patterns inherently.

5.2.8 Results

1. **Comparison with previous techniques**
   This section showcases the output values from the signal-to-signal translation for this technique. The multi modal generation is made through the Encoder, F_{xy}, and the decoder, $G_y : G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$. This technique helps generate infinite output $\hat{\mathbf{y}}$ closer to tarted signal $\mathbf{y}$ in the data sets. View results in the Figure 5.10.
   This technique performs significantly in reconstructing broadband, filtered, and hybrid signals. This architecture, as illustrated in Figure 5.9, achieved scores that surpass the SeismoALICE while ensuring the mapping one to many and maintaining a plausible variability.

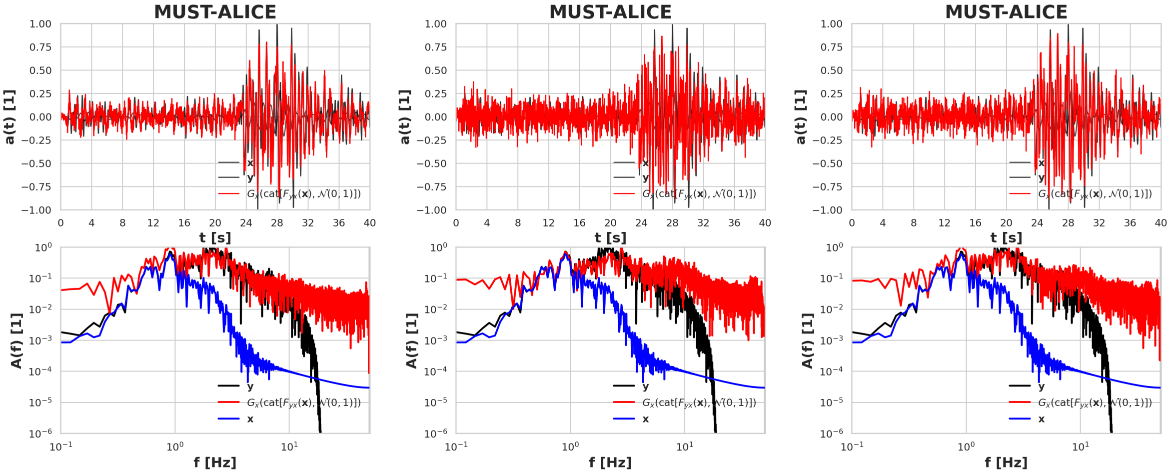
2. **Interpretability of the latent space**
   To make the common part reflect information content in the range [0-1]Hz, we thus made the information content equal in both ground motion and physic-based data. It is visually shown in Figure 5.11a and Figure 5.11b that these figures intentionally restricted



(a) Variant 1 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.

(b) Variant 2 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.

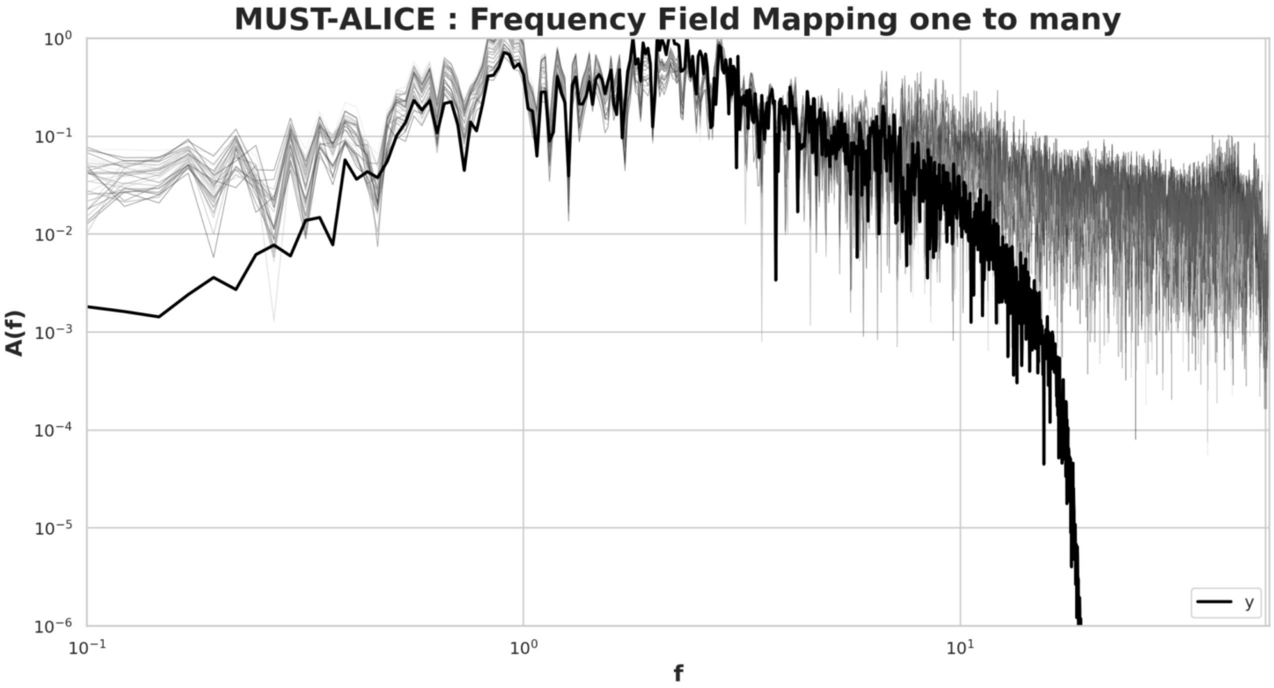
(c) Variant 3 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.



(d) Variant 4 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.

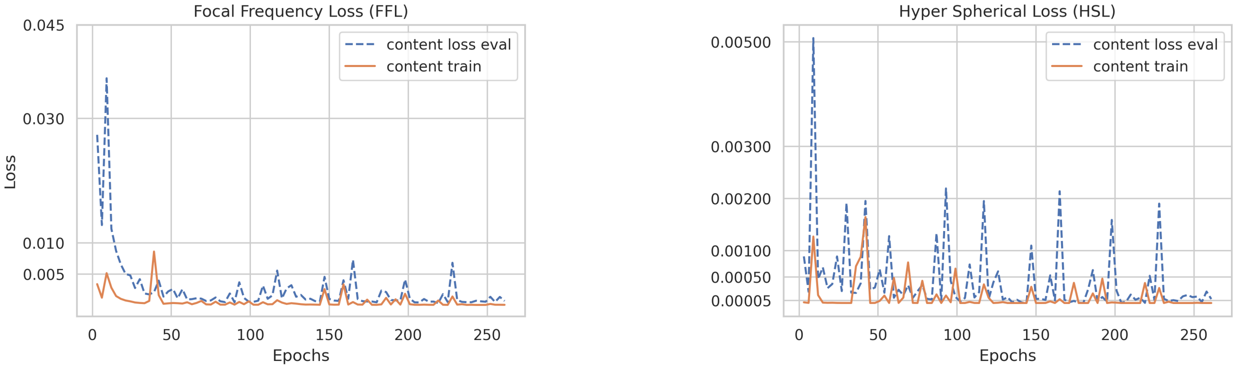
(e) Variant 5 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.

(f) Variant 6 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$, the targeted signal is $\mathbf{y}$.



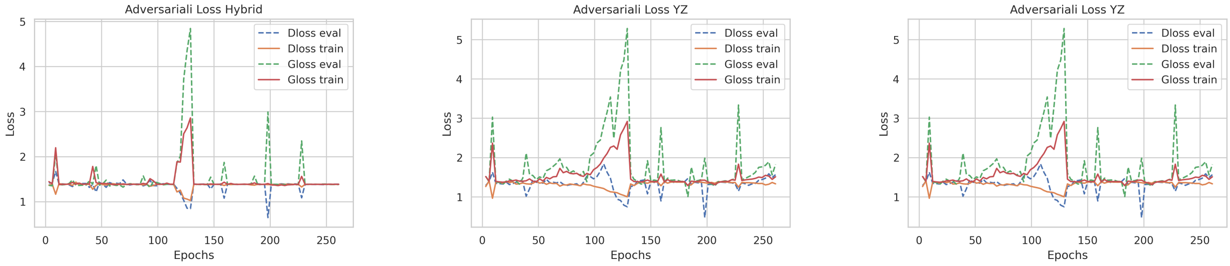
(g) Observation of the quality of the generation in the frequency field

Figure 5.10: Example of result for the Multi-modal Signal Translation



(a) Focal Frequency Loss of the content (b) Hyper Spherical Loss for content

Figure 5.11: Here is the observation of the convergence of the content part. The Focal Frequency (FFL) and the Hyper Spherical Loss (HSL) force the content part to be the same. The optimal values are close to zeros.



(a) Adversarial Loss on Reconstruction of Hybrid (b) Adversarial Loss on Reconstruction of PBS (c) Adversarial Loss on the Reconstruction of Broadband

Figure 5.12: Adversarial Loss for the Whole training of MUST technique

SeismoALICE's alignment of its content part to make sure consistency is maintained across both parts of the data. Therefore, this slight difference in the content loss helps push learning towards uniformity, guiding the model over shared info from 0 to 1 Hz. The corresponding modules are also time-synchronized within SeismoALICE.

3. **Balanced Training**
Therefore, adversarial reconstruction loss plays a role in properly updating the network during the training process. We use the same optimization technique Adam as the previous framework. The entire training follows TTUR 's dynamic with a learning rate of $1e-4$ for the Generators and a learning rate of $1e-5$ for the discriminator. Both generators and discriminators used spectral normalization.
In figure 5.12, we have investigated and presented the results for adversarial loss in discriminators and generators. The training was stopped when the gradient norms of discriminators and generators converged to zero. This training is performed on 4 GPU A100s over 20h period.

5.3 Conclusion

Through the Multimodal generation of hybrid generation quality of the UNet and SeismoALICE's variational generation, we came up with this variant inspired by BiCycleGAN.

This architecture fuses the strength of *VAE-GAN*and cLR-GAN. The former ensures that the hybrid generation's quality is upheld and that the variational aspect of the hybrid generation comes into play, whereas the latter measures this conditioned generation.

Nevertheless, further investigations and various experiments have shown that in some cases BiCycleGAN introduces irrelevant artifacts or noise as solutions. Consequently, improvements in reconstruction quality required by BicycleGAN are made at the cost of sacrificing the interpretability that SeismoALICE has over it.

In comparison, latent space design for ALICE has been found to degrade relevant information; therefore, it needs to be more comprehensible to human beings even though we improved on its interpretability. As such, our whole architecture design seems to lend itself poorly to joint information sharing.

These problems are addressed using new approaches, such as exploring a MUNIT-based framework, MUST, for better one-to-many mapping. The compressed information and uncompressed information should be simultaneously adjusted. Good results were obtained when utilizing adversarial loss alongside our custom minimization approach explicitly customized for time series data. Consequently, versatile and resistant representations that encompass content (uncompressed information) and style (compressed information) have been introduced into his representations. In other words, such architecture can better capture low-frequency components while preserving a strong representation of hybrid data.

Thus far, our exploration across different architectures has helped us accept both the potentialities offered by BicycleGAN and its constraints, especially concerning SeismoALICE. Our inquiries direct us toward the MUST framework; potential solutions are available through their investigation. This can only be done by continuous research work being tested out in practical terms.

Chapter 6

Case Study in Earthquake Engineering

> *"One path to human-level AI uses mathematical logic to formalize common-sense knowledge in such a way that common-sense problems can be solved by logical reasoning."*
>
> John McCarthy, Artificial Intelligence, Logic and Formalization common sens, 1990

6.1 Teil

The year 2022 has seen the publication of a paper on Physics-Based Simulations (PBS) for seismic activity in the Le Teil region by Lehmann *et al.* [322]. The earthquake consideration is moderate, with moment magnitude $M_w = 4.9$. It took place at a shallow depth of about one kilometer in an area with few geophysical measurements. The simulations were implemented using a high-fidelity wave propagation code that allows for numerical simulations of the Le Teil earthquake within the high uncertainty framework. In this investigation, various seismic sources and geological setups are considered to gain insights into seismicity in the region.
Profile of velocity and 3D model is presented in Figure 6.2. The Le Teil earthquake was recorded across 22 stations (Figure 6.1) within a 70 km radius from the fault. This study focuses explicitly on stations outside the sedimentary basin, such as OGDF, OGCB, and CRU1, as the absence of these stations from our models could potentially impact synthetic ground motions within the basin. Numerical simulations were conducted using SEM3D, a High-Performance Computing wave propagation code based on the Spectral Element Method. SEM3D exhibits weak scalability properties, particularly between 0 and 10,000 MPI processes. It has been widely utilized for simulating past earthquakes and evaluating the seismic response of nuclear sites and urban areas. The computational domain covered 80 km × 92 km × 79 km and was discretized on a hexahedral mesh with 18.3 million elements. With a minimum S-wave velocity of 2180 m/s and 5 Gauss-Lobatto-Legendre (GLL) points per element, this mesh enabled wave propagation up to 5 Hz. Simulations were executed on 2048 cores of AMD Milan @2.45 GHz (AVX2), hosted by the Très Grand Centre de Calcul (TGCC, France). The computational power available facilitated simulations that spanned 61,440 CPU hours for simulating a 60-second signal. Our objective is to assess the performance of our trained model (Pix2Pix,

SeismoALICE(I and II), BiCycleGAN, and MUST). We will use recorded seismograms and synthetic signals generated for the Le Teil. This evaluation judges how well our various models generalize to unseen data and validate our assumptions. It is noteworthy to tell the reader that we did not fine-tune our model to the Teil data set. Many captors were used, but according to Lehmann, only three were satisfying (OCGB, OGCD, and CRU1). The following sections will present the results for one seismogram, CRU1, but the reader can consult the Annexe for the other captors.

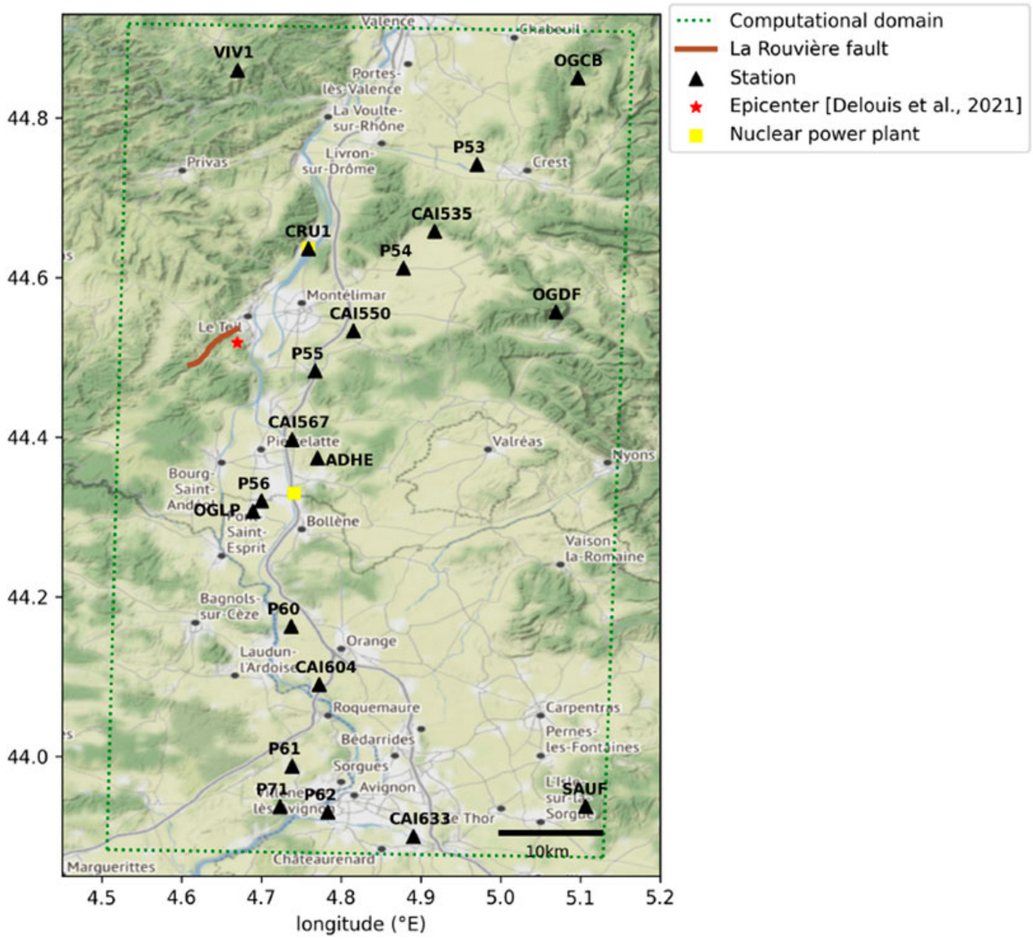
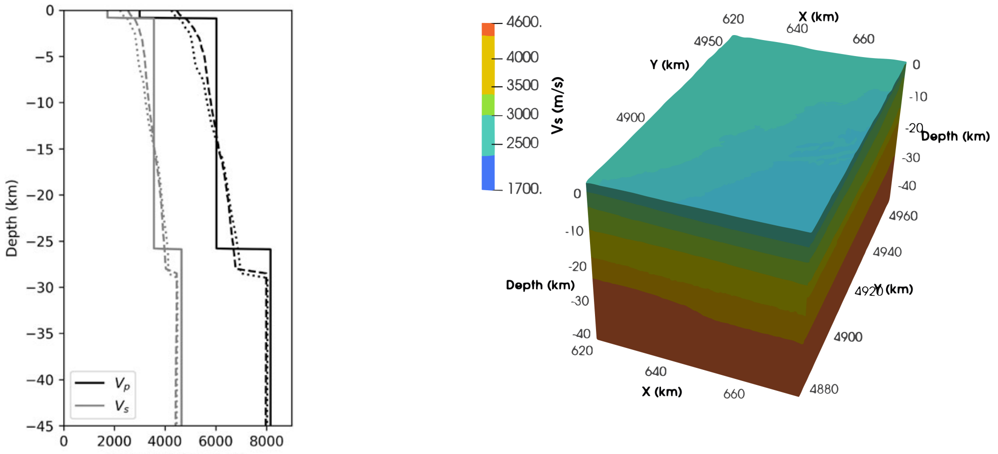


Figure 6.1: Source [322]. *"Map of the region affected by the 2019 Le Teil earthquake, in South-Eastern France. The computational domain considered in this paper is indicated with the dotted box. Velocimeters and accelerometers are shown with black triangles".*

6.1.1 Pix2Pix

We employed our UNet architecture, pre-trained on the STEAD dataset to test this effectiveness in a different dataset to predict the ground motions—the result of the physics-based simulation of Teil. Figure 6.3 depicted the outcomes of these applications, especially for the results of captor CRU1 in the three directions: North-South (NS), East-West (EW), and Up-Down (UD) (View in Annex the results for captor OGDF Figure A.6 and captor OGCD Figure A.5).

In our methodology, the PBS serves as input to the neural network, which generates a synthetic signal. The resulting synthetic signal is then visually plotted for comparison, showcasing the frequency information predictions. This comprehensive analysis provides insights into the accuracy and characteristics of the predicted ground motions.

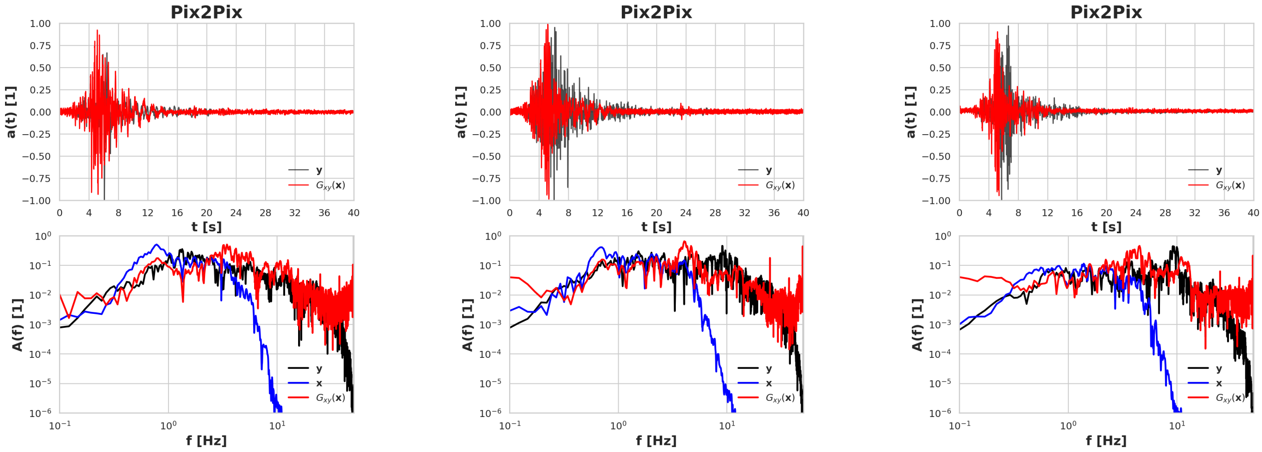


(a)

(b)

Figure 6.2: We present the geology and the profile Geology of the study area. In (a) "VP (black) and VS (grey) velocity profiles for the 1D geological model (continuous line), 3D geological model in station VIVF (dashed line), and station SAUF (dotted line).". In (b) "3D geological model for S-wave, original model". Source [322].

It is noteworthy that the synthetic signal exhibits a slightly higher amplitude than the original signal if we compare the signal in the time domain. Despite this difference, the quality of the predictions remains within the bounds of plausibility. Crucially, the model successfully captures the fundamental characteristics of seismic events, allowing for reliable representations of earthquake-related features in the predictions.



(a) NS

(b) EW

(c) UD

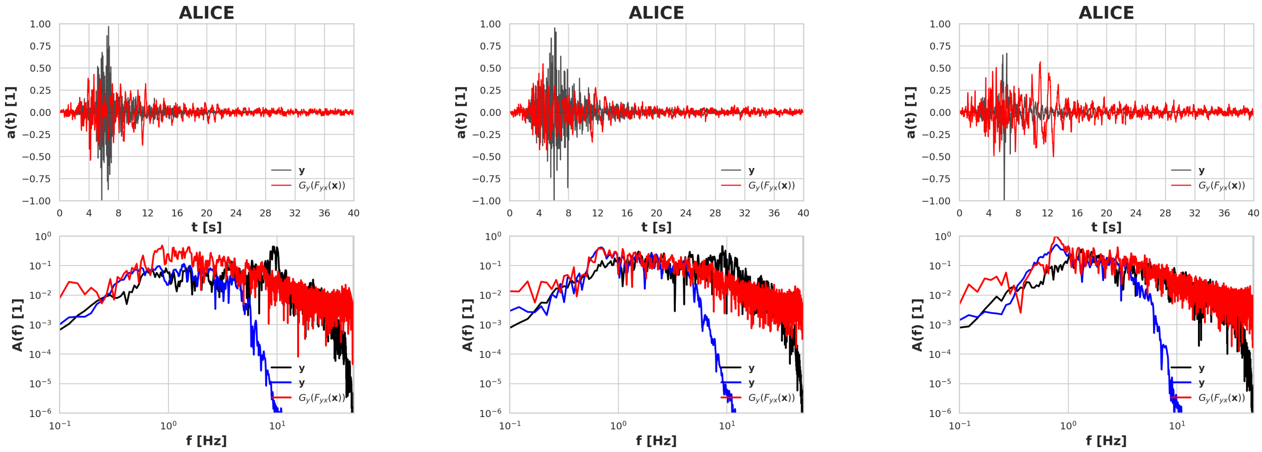
Figure 6.3: Result for for the seismogram CRU1 for the simulation of Teil

6.1.2 SeismoALICE with Shared Latent Space

To test the pre-trained encoder and decoder (F_{xy} and G_y) on our STEAD dataset, we focused specifically on results from captor CRU1. Figure 6.4 shows the outcomes for the shared latent space for seismogram CRU1.

Our approach is first to encode physics-based data into 256 values using a pre-trained encoder

(F_{xy}), then use a broadband decoder to generate synthetic data: $G_y(F_{xy}(\mathbf{x}))$. But it's clear that Pix2Pix performs better than our shared-space model regarding time-domain accuracy. The frequency domain also shows this same pattern. The precision that Pix2Pix offers seems too good for our shared-space model to match.



(a) NS (b) EW (c) UD

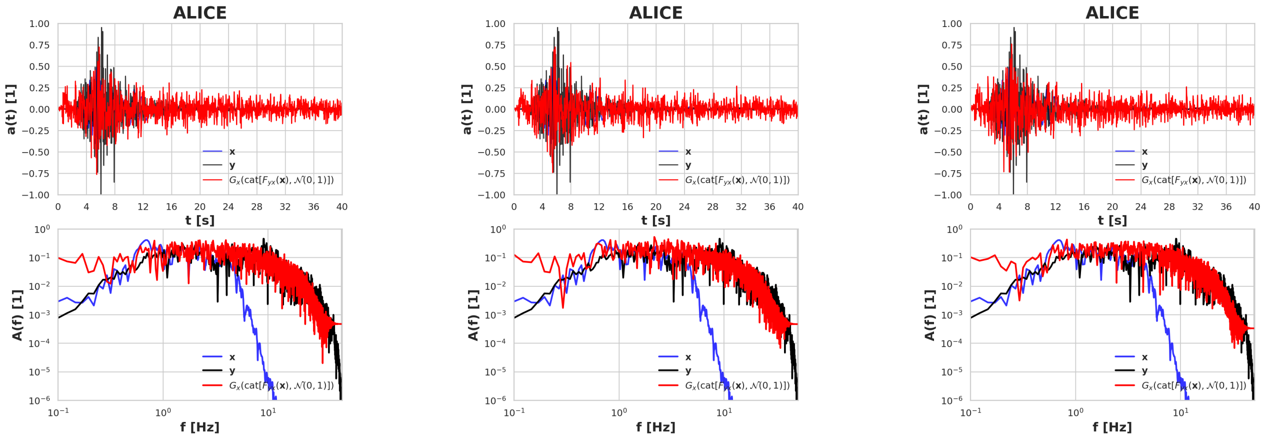
Figure 6.4: We present the hybrid generation for ALICE Shared latent space for the seismogram CRU11

6.1.3 SeismoALICE with Factorized Latent Space

We conducted a thorough investigation into the efficacy of both one-to-one mappings and one-to-many mappings by applying them to the seismogram CRU1 (results for captor OGCB in Figure A.3 and captor OGDF in Figure A.4). The one-to-many mapping is illustrated in Figure 6.5. We observe a comprehensive representation of the targeted signal. We observe the number of variants of the same signal in the frequency fields. Notice that the phase alignment is respected.

6.1.4 BiCycleGAN

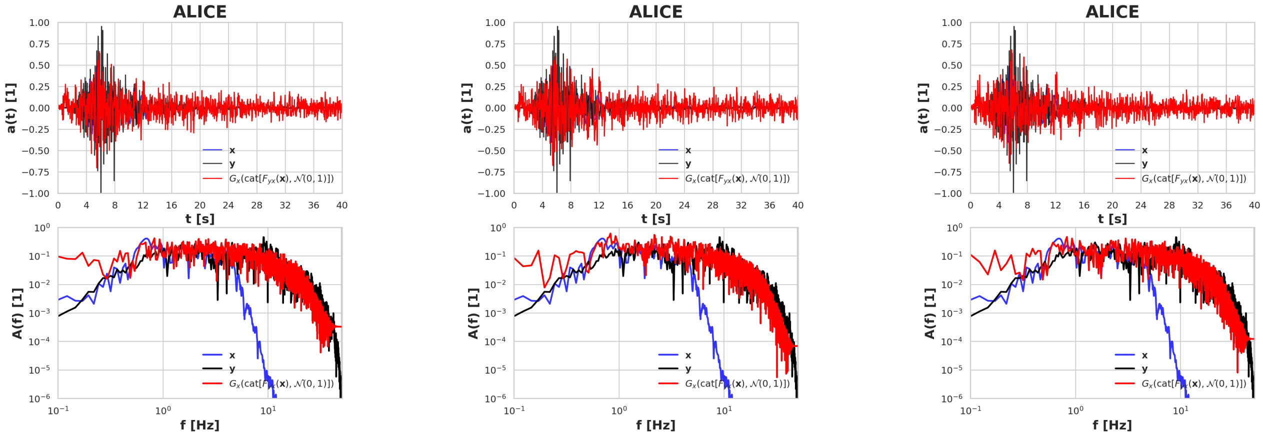
Figure 6.7 and Figure 6.6 show the result of one-to-many mapping for BiCycleGAN. The decoder, G_y, takes two inputs. The signal from the PBS sensor $\mathbf{x}$, and what can be considered to be the encoding of the broadband signal or its equivalent, a Gaussian distribution, of the same type. The previous BiCycleGAN study revealed that full latent space injection works best for one-to-one and multimodal mapping. In fact, under real conditions, the corresponding $\mathbf{y}$ values will not be known, but for model performance research, we have taken the liberty of outputting $G_y(\mathbf{x}, F_{xy}(\mathbf{y}))$ (See Figure 6.6). We have also provided in Figure 5.4, the answer for $G_y(\mathbf{x}, \mathbf{N}(0, I))$. $G_y(\mathbf{x}, \mathbf{N}(0, I))$, which means we can see how well the reconstructions are done in this architecture. Also, observations for the one-to-one mapping show how difficult it is to estimate the low frequencies [0,1] in the frequency domain. This issue is a numerical limitation of our architecture at the moment. On the other hand, we can see that the amplitudes are very close for higher frequencies. On the other hand, the one-to-many mapping produces plausible and consistent results with the seismic record from the CRU1 sensor.



(a) Variant 1 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$

(b) Variant 2 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$

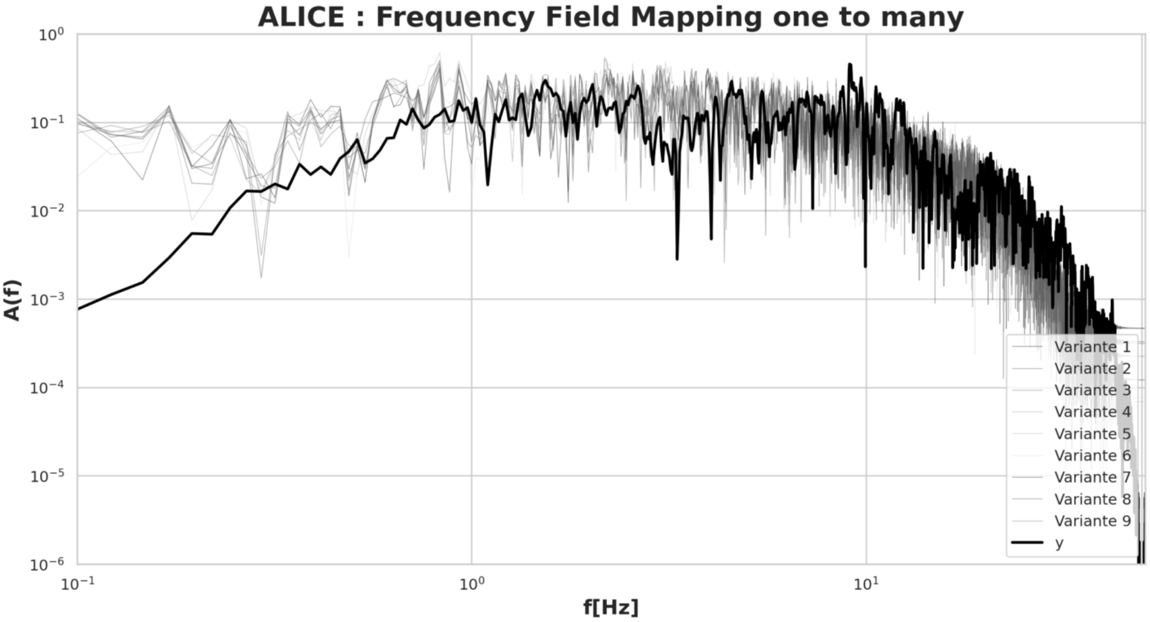
(c) Variant 3 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$



(d) Variant 4 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$

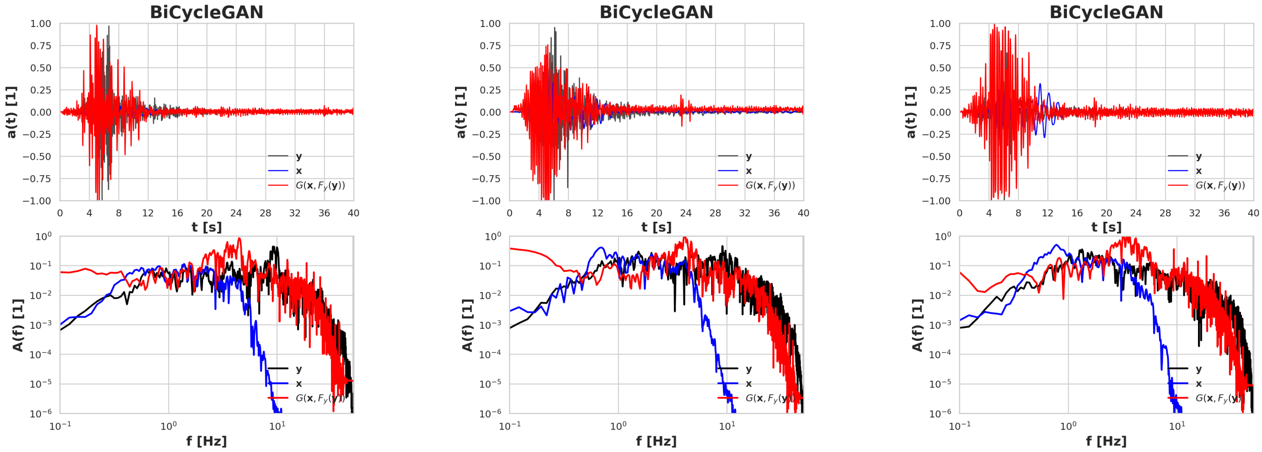
(e) Variant 5 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$

(f) Variant 6 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0, I))$



(g) Frequency Filed of the different variants from Splitted latent space of ALICE

Figure 6.5: SeismoALICE with splitted latent space. Seismogram CRU1 is used. We have presented the multiple variant for direction UD, as example



(a) (b) (c)

Figure 6.6: We present the result for the BiCycleGAN. Mapping one to one

6.1.5 MUST

The prediction with MUST-ALICE involves calculating the content of the PBS signal. The style part is replaced with a Gaussian noise, $\mathcal{N}(0, I)$. Therefore the targeted version is, $G_y F^c_{xy}(\mathbf{x}, \mathcal{N}(0, I)))$. In the time domain, the network fails to reproduce the appropriate amplitude. However, the synthetic signal remains close to the targeted acceleration in the frequency field. View Figure 6.8.

6.2 Comparison Between Different Models

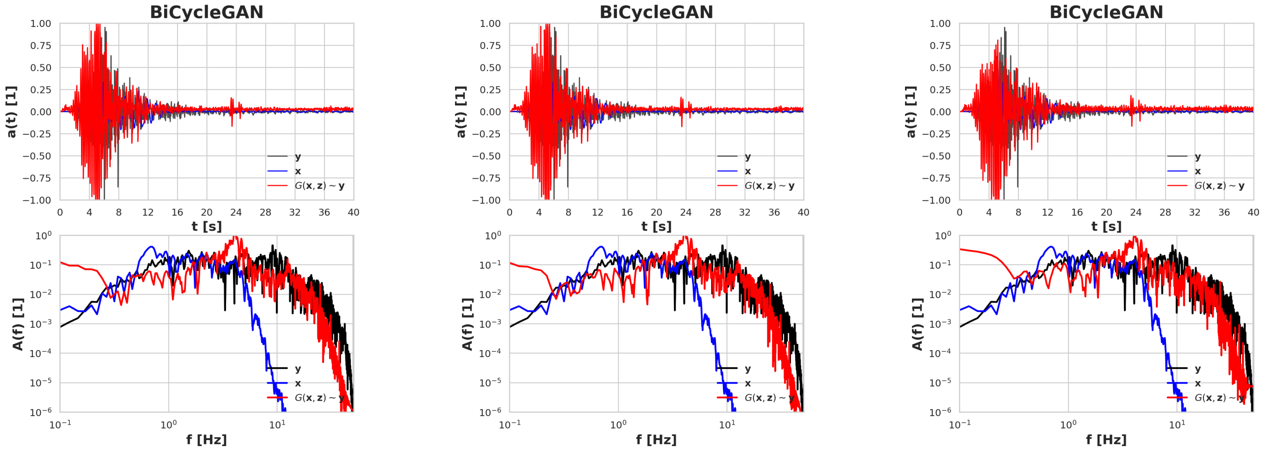
The results of the Teil dataset, especially when CRU1 captor is used, validate the efficacy of our diverse architectures. The aim was to improve physics-based simulations with machine learning. In this instance, the input was an acceleration signal ranging from 0 to 1 Hertz. However, the synthetic output signal reaches over 30 Hz and thus captures a wide range of comprehensive data that is impossible using traditional methods.

1. **Frequency accuracy**. When assessing frequency accuracy, it is noteworthy that none of the models precisely replicate amplitudes within the 0 to 1 Hz range. However, among the models considered, MUST-ALICE demonstrates the lowest deviation from the original signal.

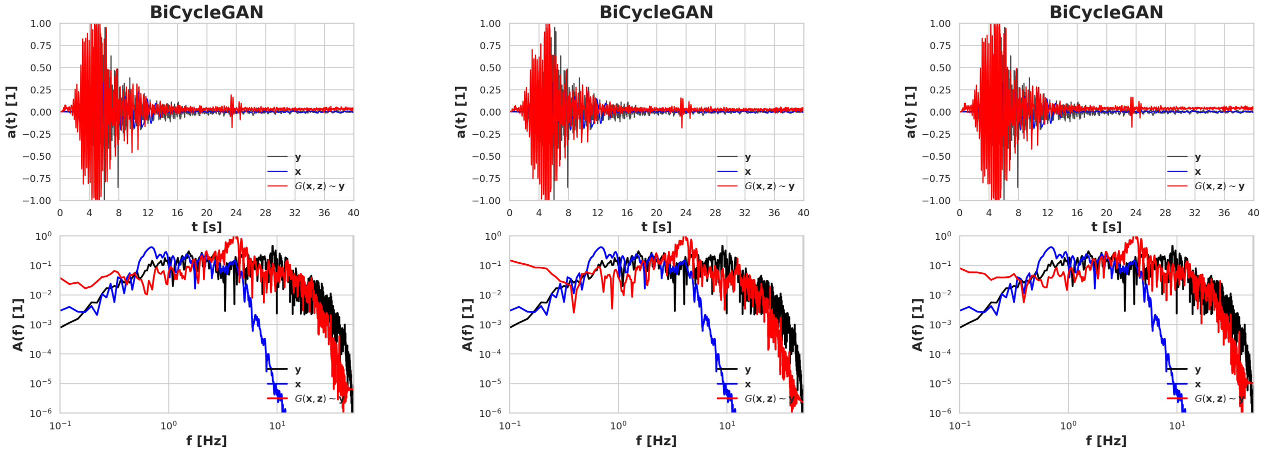
2. **Amplitude Fidelity**. Analysis of amplitude fidelity reveals that both variants of ALICE (shared and factorized latent space) better adhere to the amplitude values of the signals. This suggests that ALICE architectures are more effective in preserving amplitude characteristics.

3. **Temporal precision**. Across different architectures, there is generally good adherence to arrival times. However, ALICE with factorized latent space introduces more noise, and MUST-ALICE exhibits a slight delay. Enveloping the amplitude differences mitigates the significance of these variations.

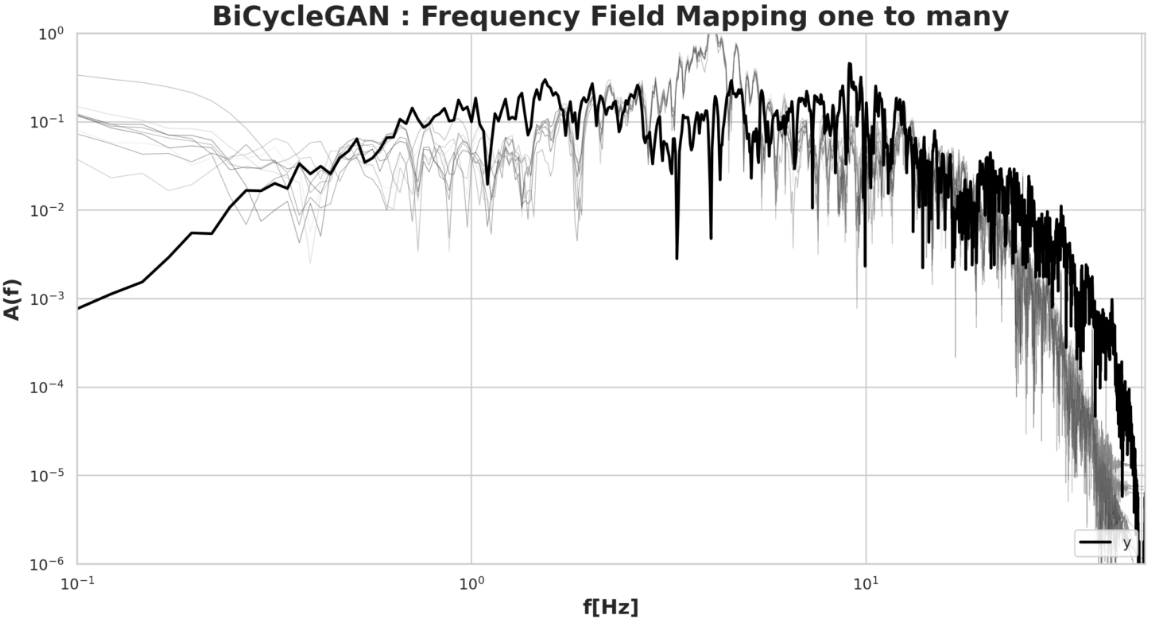
4. **One-to-one vs one-to-many mapping** One-to-one mapping produces plausible signals, while one-to-many mapping offers multiple valid solutions for the same input. Frequency domain analysis helps understand how variants stay close to the original signals.



(a) Variant 1 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$ (b) Variant 2 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$ (c) Variant 3 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$

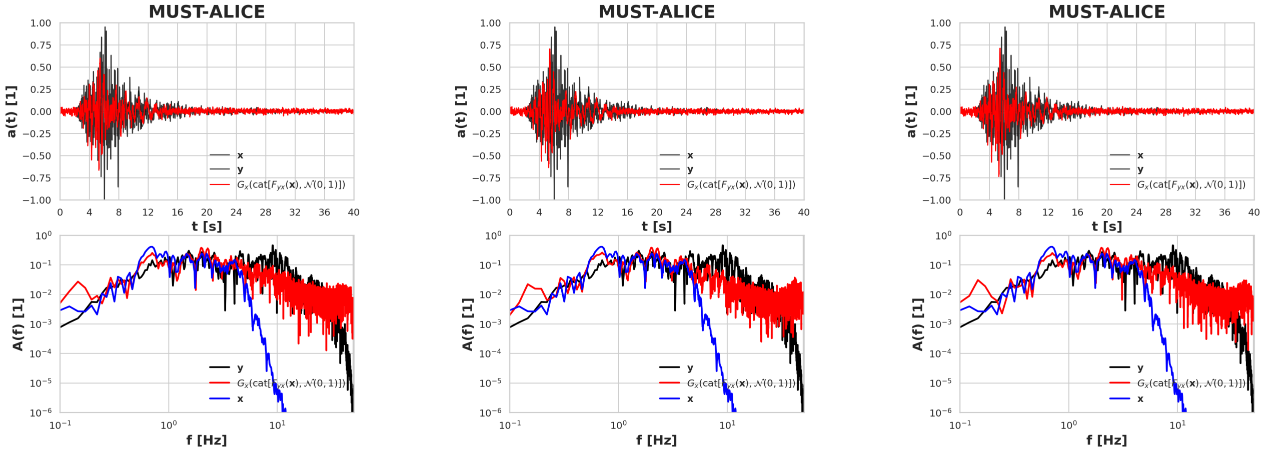


(d) Variant 4 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$ (e) Variant 5 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$ (f) Variant 6 of $G_y(\mathbf{x}, \mathcal{N}(0, I))$



(g) Frequency Filed of the different variants from splited latent space of BiCylceGAN

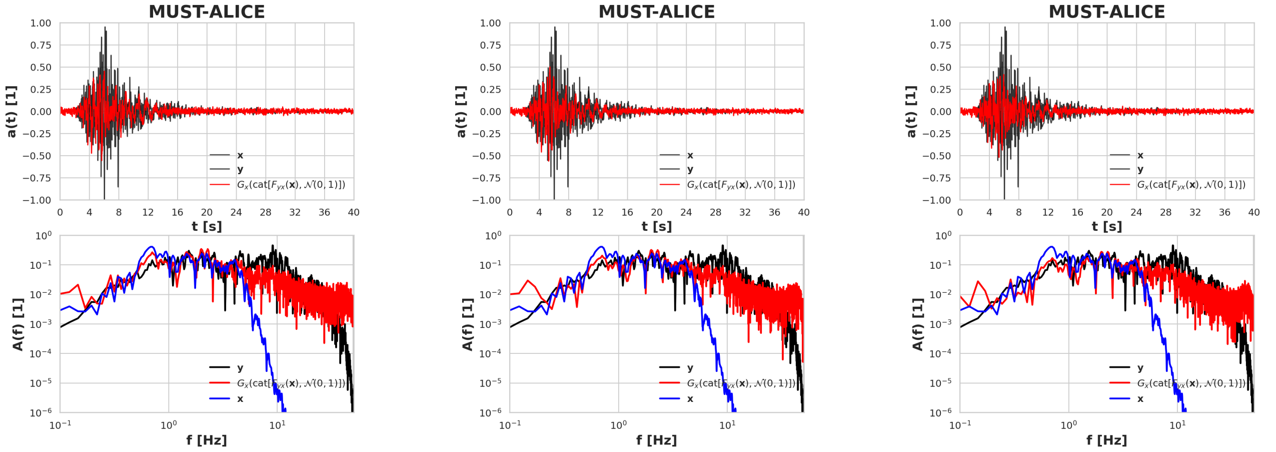
Figure 6.7: BiCycleGAN with split latent space. Seismogram CRU1 is used. We have presented the multiple variants for direction UD as an example.



(a) Variant 1 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$

(b) Variant 2 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$

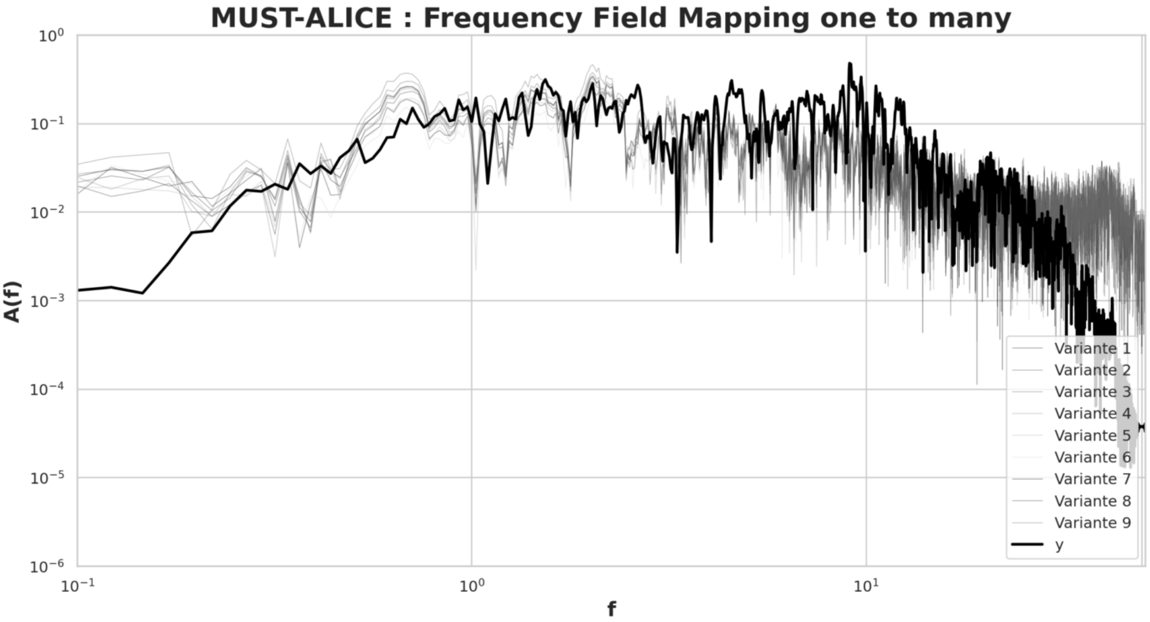
(c) Variant 3 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$



(d) Variant 4 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$

(e) Variant 5 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$

(f) Variant 6 of $G_y(F^c_{xy}(\mathbf{x}), \mathcal{N}(0, I))$



(g) Frequency Filed of the different variants from MUST

Figure 6.8: *MUST* with splitted latent space. Seismogram CRU1 is used. We have presented the multiple variants for direction UD, as an example

5. **Computation Efficiency**. In terms of computational efficiency, MUST-ALICE, ALICE, and their variants are computationally intensive, requiring approximately ten times the number of parameters compared to BicycleGAN and Pix2Pix. Training complexity and model weight contribute to the resource-intensive nature of MUST-ALICE and ALICE.

6. **Plausibility and Realism** The overall method yields plausible outputs, with styles providing closer outputs, except for Pix2Pix. Pix2Pix is sensitive to the input form, introducing unrealistic information into the signal.

6.3 Technical Improvement

6.3.1 Difficulty of Extraction of Data

In the STEAD database, all the signals are in the form of .h5py files. These files contain 3D signals with a total of 6000 time steps. However, extracting and processing this data on a single CPU is slow. We need more data to improve our models, but only a limited amount of sequential extraction can be applied to this problem before it becomes too slow. So we need to find a better way of solving this problem. A more efficient approach was needed, so we developed an algorithm to process the extraction in parallel.

6.3.2 Parallel Extraction of Data

Our solution leverages the organizational structure of the STEAD database, which utilizes pandas data frames. This unique organization facilitates the parallel extraction of the entire dataset. Specifically, our goal is to extract 128,000 3D signals. Through the application of our algorithm, a task that would originally demand over 20 days for completion can now be accomplished in just 20 minutes, utilizing the processing power of 80 CPU cores(View Figure 6.10). The fundamental code implementing this process can be found in the Appendices, specifically in Section A.3.1 and Section A.3.2. Figure 6.9 illustrates the algorithm's performance to extract data.

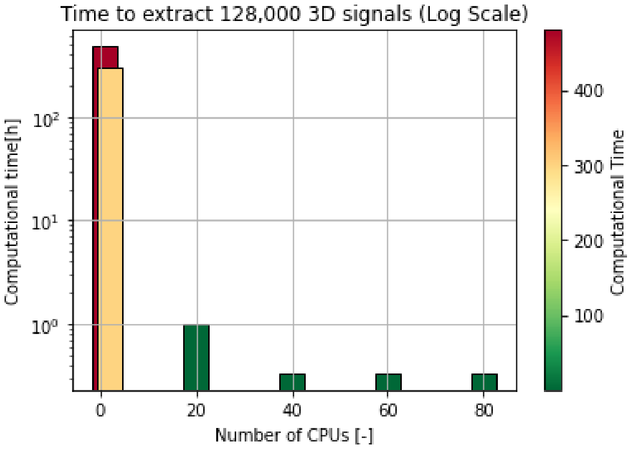


Figure 6.9: Computational time to extract 128,000 3D signal per number of CPUs. This estimation times. Clearly explain the performance of our algorithm.

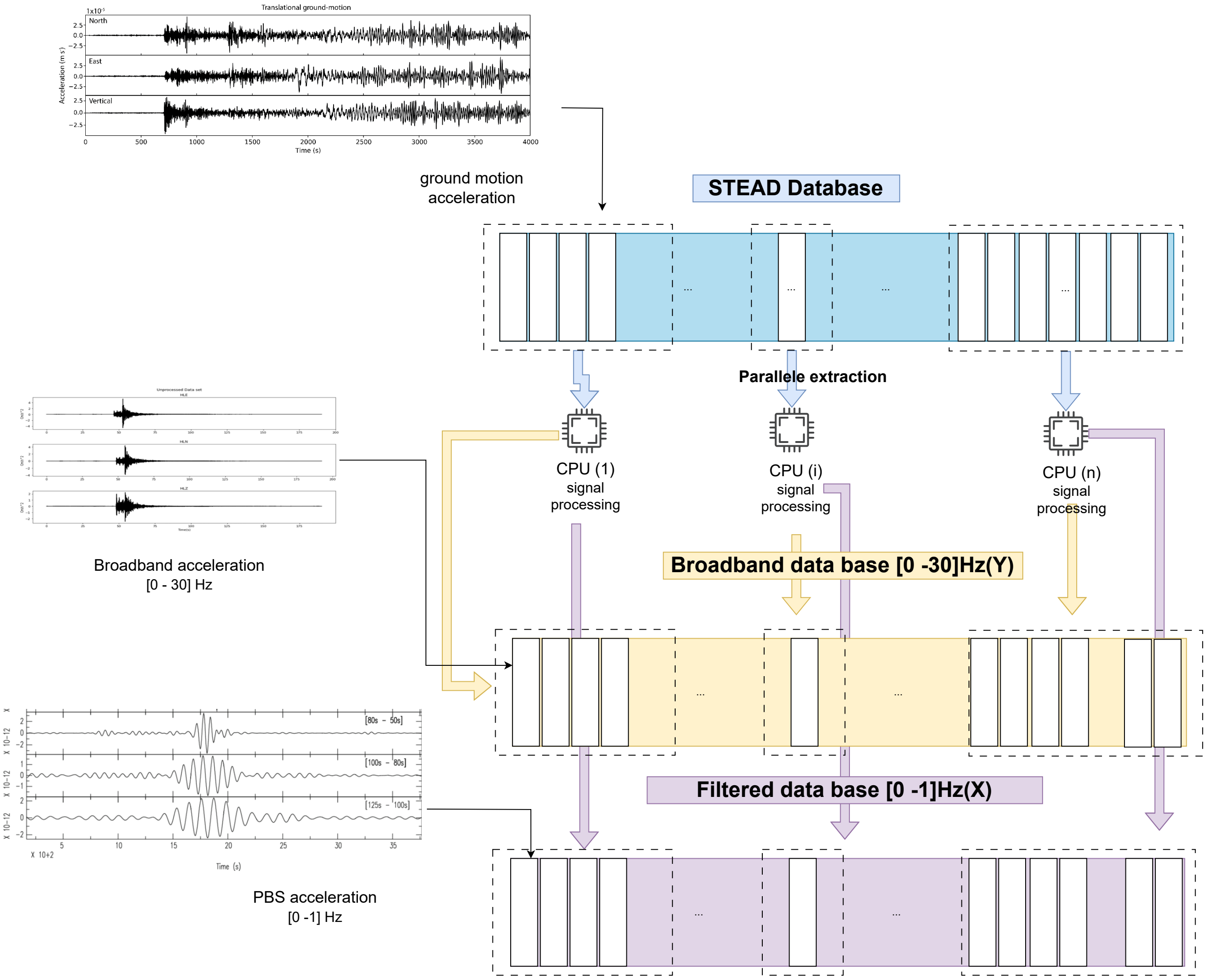


Figure 6.10: Explanation of parallel processing of STEAD signal. We divide the dataset into different chunks. Each chunk contains is allocated to a CPU. An amount of 3D accelerogram. And is processed and output broadband, PBS, and the metadata

6.4 Conclusion

In the previous chapters, we have tested 5 different architectures. These architectures have been tested on the subset of the Teil. The encoders and decoders have not trained another time on the sub-set of the Teil. Among the various signals, we choose to present the result for the captor on the nuclear powerplant. As input, we use the result simulation of Lehmann *et al.*, 2022 [322]. The different trained models on the STEAD database have proven that it is possible to transform physic-based signal (PBS) to ground motion data. We transform acceleration where the values in the frequency domain do not exceed 1Hz to output a synthetic signal rich in frequency up to 30 Hz. The performance of the different models could be summarized as follows:

- **Signal-to-signal Translation** has the advantage of producing realistic output consistent with seismic recording but is highly sensitive to noise.
- **SeismoALICE with shared latent space** achieves better ground motion predictions than Signal-to-Signal Translation. However, it is constrained to one-to-one mappings.

- **SeismoALICE with split latent space** could produce plausible one-to-many mappings but introduce noise in this prediction.
- **MUST-ALICE**, also produce realistic one-to-many mapping, clearly predict the ground motion. However, this latter reduces the noise in his prediction.

Our extensive testing on the Teil simulations highlights the effectiveness of these predictive models for future earthquake engineering tasks.

One of the limitations of training complex models is ensuring they are fed with sufficient data, particularly for intricate tasks. To address this challenge, we have investigated different proper training methods with a reasonable amount of data. Given the task's difficulty, we found that more than 100,000 3D signals were essential for robust model performance. Our investigation has proven that classic extraction cannot train 128,000 3D signals. We have, therefore, solved the technical limitation by developing a parallel extraction algorithm. The classical time to proceed with one CPU would take 480 hours (20 days); this time is reduced with 80 CPUs parallel to 0.33h (20 minutes). This massive point of solution has helped us solve overfitting issues and improve generalization and other difficulties in adversarial training.

Finally, Our findings have demonstrated the remarkable potential of our method to transform low-frequency PBS into high-fidelity ground motion data, opening up new avenues for seismic modeling and analysis. The development of our parallel extraction algorithm has further enhanced the practicality and efficiency of our approach, paving the way for its broader application in earthquake engineering.

Chapter 7

Conclusions and Perspectives

> *"We are on the threshold of an era that will be strongly influenced, and quite possibly dominated, by intelligent problem-solving machines."*
>
> — Marvin Minsky, 1961

7.1 Mapping physics-based into broadband signals

Faced with the limits of physics-based simulation to predict the output simulations with high fidelity, we have explored in this manuscript how machine learning could help fill the gap left by this strategy at high frequency. An adversarial generative framework was chosen to tackle the problem and to improve the numerical simulations, assumed accurate enough in the 0-1 Hz frequency range, to render realistic broadband (0-30 Hz frequency band) earthquake ground motion synthetics while preserving the physics simulated at low frequency. Major challenges related to time series super-resolutions have been thoroughly investigated. We have tested the most suitable neural architectures and adapted the loss functions, explicitly tailored for time series, to address various limitations arising from the original development of such image techniques. These advancements have resulted in a more robust and comprehensive exploration of signal-to-signal translation, allowing us to render multiple alternative realizations of realistic broadband earthquake accelerograms. One key finding of this work is the effectiveness of a discriminator structure incorporating a residual neural network, closely mimicking the encoder structure. This architecture resulted in stability against adversarial training, contributing to the overall success of our approach to signal-to-signal super-resolution translation. In this sense, combining Focal Frequency Loss (FFL) and Hyper Spherical Loss (HSL), outperforms traditional L1-norm or L2-norm for time series data. This combination improves scoring and demonstrates that HSL can serve a dual role as both a loss function and a metric. These findings highlight the importance of tailored architectures and cost functions in overcoming the unique challenges of time series data manipulation. Also, to achieve the signal-to-signal super-resolved translation, we tested five methods, namely Pix2Pix, SeismoALICE with shared latent space, SeismoALICE with factorized latent space, BiCyleGAN, MUST, each characterized by different advantages and limitations, that were thoroughly investigated.

With Pix2Pix, we have shown the feasibility of the direct one-to-one mapping $\mathbf{x} \xrightarrow[G_{xy}]{} \mathbf{y}$.

This mapping was achieved by training a UNet architecture, capable of transforming the PBS into a broadband realistic signal.

The novel approach introduced in SeismoALICE with shared latent space involves a single encoder, denoted as F_{xy}, which compels latent values to be interpretable, so that two different decoders, namely G_y and G_x can effectively generated both PBS and broadband record-like time series. This innovative architecture imposes specific criteria on the entire neural network, namely:

(i) broadband ground motion reconstruction $G_y(F_{xy}(\mathbf{y}))$;

(ii) physics-based simulation reconstruction $G_x(F_{xy}(\mathbf{x})$;

(iii) hybrid mapping from physics-based simulation to ground-motion-like recordings $G_y(F_x y(\mathbf{x}))$;

(iv) Hybrid mapping from ground-motion-like recordings to physics-based simulations $G_x(F_x y(\mathbf{y}))$;

(v) Unconditional generation of broadband realistic ground motion time histories $G_y(\mathbf{z}_y)$;

(vi) Unconditional generation of physics-based numerical simulations $G_x(\mathbf{z}_x)$

In the SeismoALICE with factorized latent space, we enhance the architecture in order to perform a one-to-many mapping, *i.e.* from one PBS $\mathbf{x}$ we could output an infinite number of different hybrid acceleration time histories, that resembles the targeted broadband signal but that preserve the low-frequency part of the spectrum learnt from the physics-based simulation. The whole criteria to satisfy are :

(i) broadband ground motion reconstruction, $G_y(F_{xy}^{yx}(\mathbf{y}), F_{xy}^{yy}(\mathbf{y}))$;

(ii) physics-based simulation reconstruction
$G_x(F_{xy}^{xy}(\mathbf{x}))$;

(iii) enforcing the consistency of the common features extracted from physics-based and broadband recorded signals
$F_{xy}^{xy}(\mathbf{x}) = \mathbf{z}_{xy} \sim F_{xy}^{xy}(\mathbf{y}) = z_{yx}$;

(iv) enforcing each distinct encoding to match normal distribution

(v) unconditional generation of broadband realistic ground motion time histories $G_y(\mathbf{z}_y)$;

(vi) unconditional generation of physics-based numerical simulations $G_x(\mathbf{z}_x)$;

(vii) conditional generation of many hybrid ground motion time series conditioned by low-frequency physics-based simulations
$G_y(\mathbf{x}, \mathcal{N}(0, \mathbf{I}))$

(viii) Enforcing phase alignment for one-to-many generation.

In the case of the BicycleGAN approach, tailored for signal-to-signal super-resolution translation, in addition to the UNet architecture that targets the broadband data, we introduce a feature extractor, whose output is injected into the UNet layers. Such manipulation introduces output variability and allows the network to achieve one-to-many mapping.

With MUST, we have pushed a step further the one-to-many mapping framework, towards a more intuitive interpretation of the latent variables while satisfying the following criteria:

Method	**Variational Aspect**	**PBS and Broadband reconstructions**	**Complexity of the training**	**Mapping one-to-many**	**Best Hybrid Reconstruct. Sorce EG**	**Best Hybrid Reconstruct. Score PG**
Pix2Pix	No	No	Low	No	7.45± 0.79	8.63± 0.67
SeismoALICE with shared latent space	Yes	Yes	High	No	5.95±0.78	7.73± 0.43
SeismoALICE with splitted latent space	Yes	Yes	High	Yes	5.80 ± 1.07	8.32 ± 0.46
BicycleGAN	No	No	Medium	Yes	6.87 ± 1.28	8.08 ± 1.11
MUST	No	Yes	High	Yes	6.28± 0.6	8.24± 0.53

Table 7.1: Table of review of the different architectures developed to perform signal translation.

(i) broadband ground motion reconstruction, $G_y(F^s_{xy}(\mathbf{y}, F^c_{xy}(\mathbf{y}))$;

(ii) physics-based simulation reconstruction, $G_x(F^s_{xy}(\mathbf{x}, F^c_{xy}(\mathbf{x}))$;

(iii) conditional generation of many hybrid ground motion time series conditioned by low-frequency physics-based simulations $G_y(F^s_{xy}(\mathbf{y}), F^c_{xy}(\mathbf{x})$;

(iv) enforcing consistency of each specific encoding branch through the pairs $\mathbf{x}$ and $\mathbf{y}$, $F^c_{xy}(\mathbf{y}) \sim F^c_{xy}(\mathbf{x})$;

(v) unconditional generation of broadband realistic ground motion time histories $G_y(F^c_{xy}(\mathbf{y}), \mathcal{N}(0, \mathbf{I}))$

7.2 Comparisons of Different methods

In Table 7.1, we have provided an overview of the various methods explored. The chosen criteria encompass the variational aspect, *i.e.*, assessing whether or not the method could target signals in the database from a normal distribution. The "PBS and Broadband reconstructions" criterion assesses whether the method achieved cycle consistency. The "Complexity of the training" criterion is introduced to check if the training process included adversarial loss, reconstruction loss, and regularization techniques. The "Mapping one-to-many" criterion is assessed based on the method's capability of hybrid generation, determining if the method can generate an infinite number of signals resembling the targeted broadband signal from PBS at low frequency. The final criterion is targeted as "Hybrid reconstruction", *i.e.* the envelope and phase GoF (EG and PG respectively).

The analysis of the various signal-to-signal super-resolved translation frameworks highlights the intricate nature of the task. The UNet architecture, while effective in targeting ground motion data, is highly sensitive to noise, making it challenging to obtain accurate output when imposing satisfactory variability. The BicycleGAN architecture can bypass this aspect, but could also introduce irrelevant artifacts and thus struggles in handling large signal variations. SeismoALICE with shared latent space introduces latent space interpretability but only ensures one-to-one mapping; on the contrary, SeismoALICE with factorized latent space provides a deeper insight of how the underlying probability distribution associated to earthquake time histories can be transformed into a factorized normal one. MUST allows to perform a one-to-many mapping and to capture common information present in both $\mathbf{x}$ and $\mathbf{y}$. The variability is introduced via the style branch, introducing better interpretability while preserving the quality of the reconstruction for both broadband and PBS signals. Despite that enhancement, this framework suffers from reconstruction error, because the mutual information $-I(\mathbf{x}; \mathbf{z})$ is seemingly intractable. The results of our study suggest that it is feasible to fill the information gap between 0-1 Hz and 1-30 Hz for earthquake time series, with plausible hybrid outputs.

The employed architectures facilitated this transformation, showcasing their effectiveness in handling the complexities inherent in the data and achieving the desired outcomes.

7.3 Perspectives

In future research endeavors, It could prove beneficial to investigate strategies aimed at maximizing the dependencies within data in its latent representation. The development of our architecture uses advanced design for time series. Even though the quality of the reconstruction has GOF from Good to Excellent, It could be valuable to improve Performance with advanced Transformers techniques such as Cross-Attention, Chen *et al.*, 2021[323]. Nowadays, diffusion models are popular in literature as generative models for performing one-to-many mapping. Future tests on the same dataset might be envisioned. Enhance physics-based constraints in the training scheme is paved by our research with MUST-ALICE. Another constraint, as CLIP, Patashnik *et al.*, 2021[288], could be crucial in finding more similarity within the low-frequency signal. Introducing metadata to condition the generation (V_s30, M_w, distance from the epicenter, hypo center depth) would have better control on the generation of the data. It might help interpolate in the regions where stored data is not present. This work might be seen as a continuity of the work of Florez and collaborators [88]. Testing the results in a smaller geographical area could give researchers and engineers the information they need for future site-specific and structural analyses. One improvement of this thesis is the exploitation of HSL to avoid data normalization. If, as an assumption, we normalize the broadband data, which makes the PGA. The issues when manipulating non-normalized data might be avoided with this loss. Our results on the earthquake of the Teil could serve for future tasks. Fine-tuning the model with site-specific datasets will significantly improve the predictions and the GOF.

Appendix A

Experiments

A.1 Experiments ALICE

A.1.1 Simple Strategy Broadband

In this section we present the resul of the training of broadband signal using Hinge loss as adversarial loss. Figure A.1.

A.1.2 Unified Strategy

We present results for the architecture Conformer+AdaIN+UNet. See Figure A.2.

A.2 Experiments Pix2Pix

A.2.1 Test on Captor of Teils

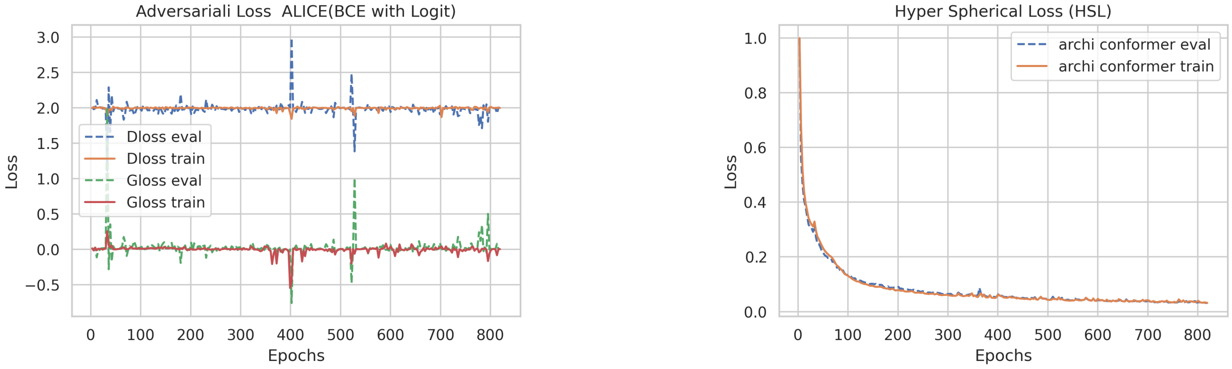
The test about Teils in the Figure A.5 and Figure A.6

A.3 Database Files

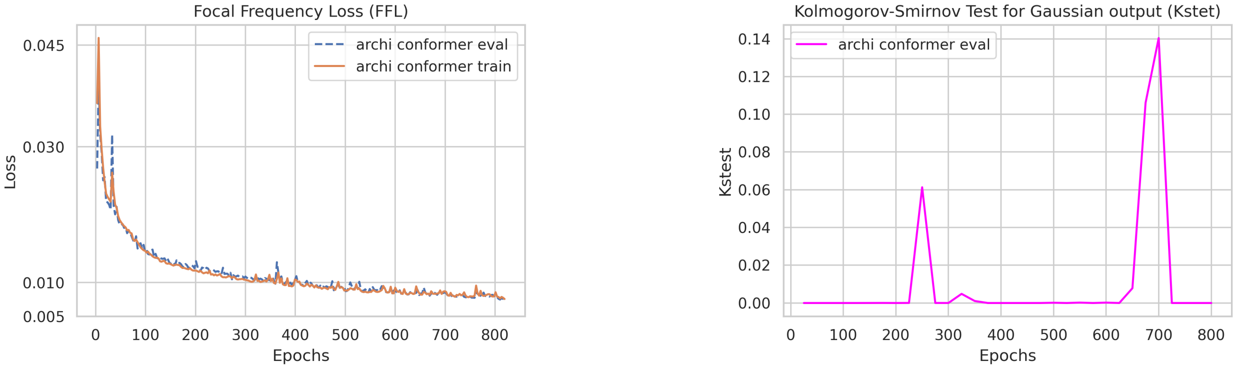
The code for the parallelization the extraction of the STEAD database

A.3.1 STEAD Extractions

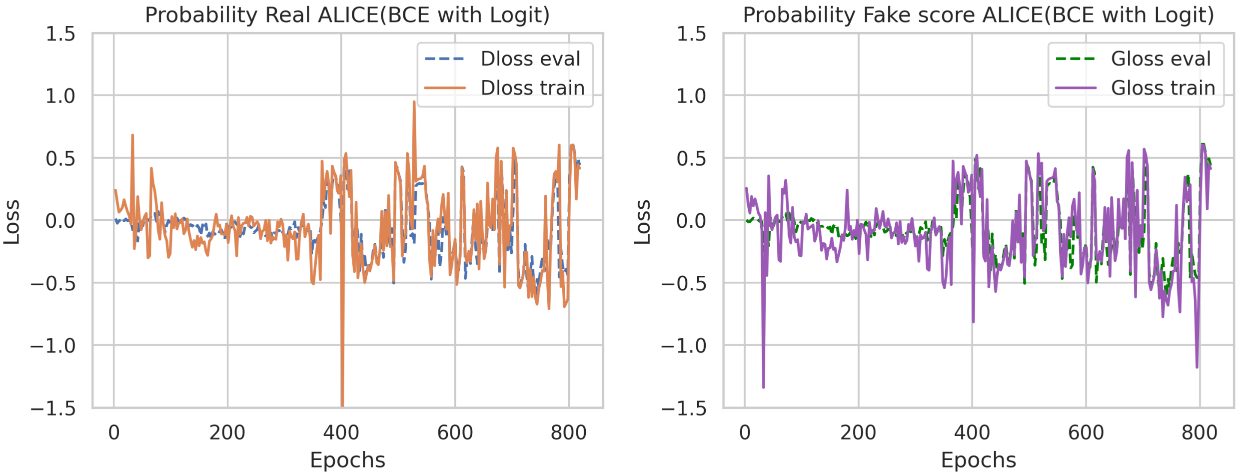
```
import os
import h5py
from tqdm import tqdm, trange
import torch
import pandas as pd
import numpy as np
import multiprocessing
import concurrent.futures
from configuration import app
import scipy.signal as signal
from tools.generate_noise import lowpass_biquad
from scipy.signal import detrend, windows

class STEADExtractorDataset():
```



(a) Adversarial solution for ALICE (b) Hyper Spherical Loss (HPS)



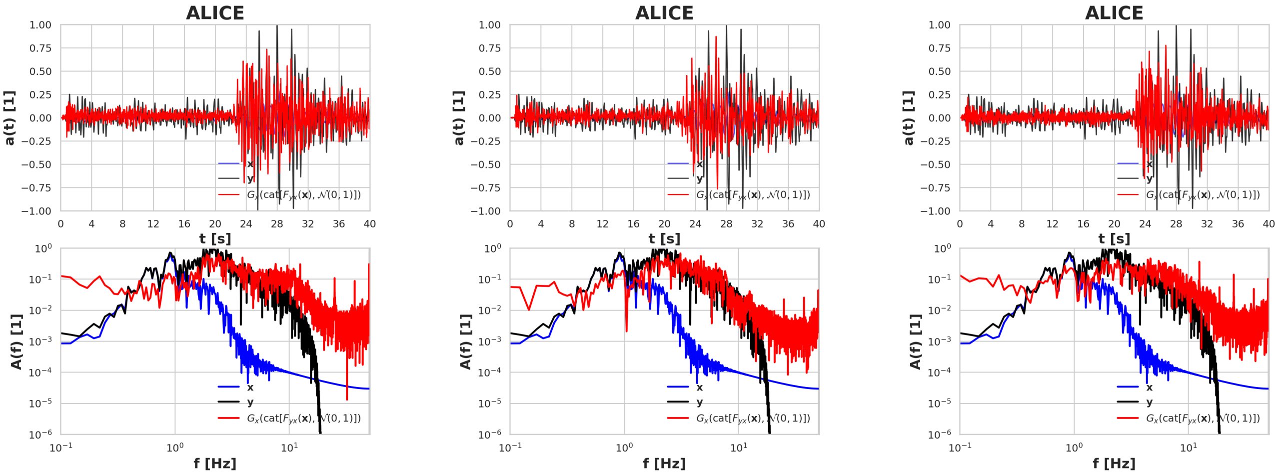
(c) Focal Frequency Loss (FFL) (d) Kolmogorov-Smirnov Test



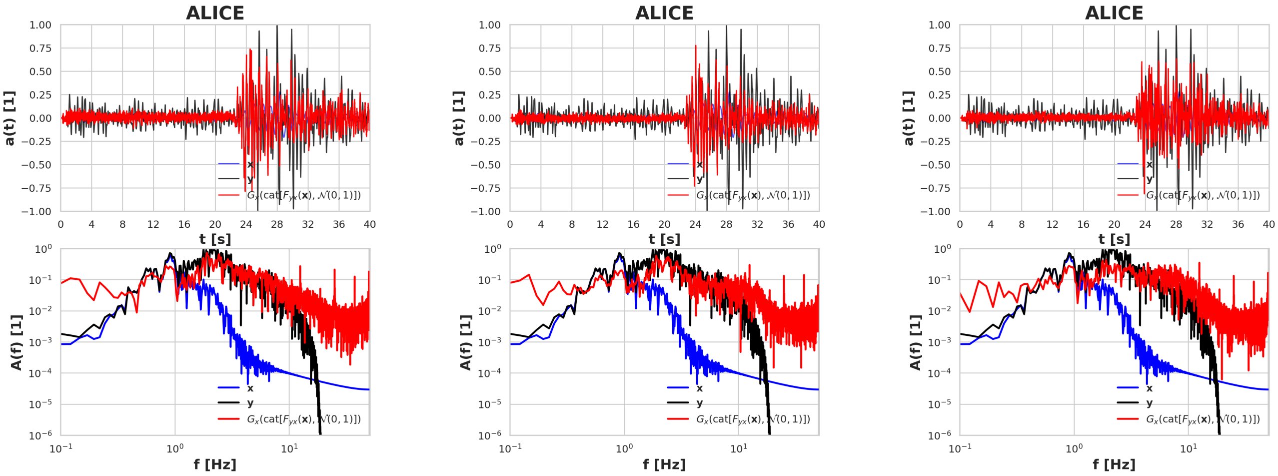
(e) Presenting the score for real and for fake probability distribution captured during the training.

Figure A.1: Adversarial Solution Training. In (a) represent the Adversarial loss, the optimal solution is 2 the loss remain close to the optimal solution due to `TTUR` dynamic training. In (b) we present the `HSL` of quality of reconstruction, the optimal solution is getting close to 0, which is the norma behavior. In (c) We give the FFL loss. We found a small overfitting of the loss, the theorical optimal solutions is 0. In (d) We present the *Kolmogorov-Smirnov* Test to evaluate the quality of the targeted distribution, $F_y(\mathbf{y})$ that should be Gaussian. The ordinate value is the probability of the distribution being a Gaussian distribution.

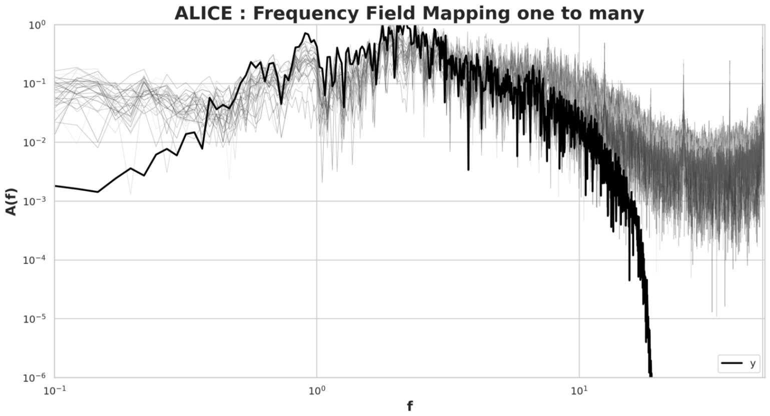
```
    def __init__(self, opt):
        super(STEADExtractorDataset, self).__init__()
```



(a) Variant 1 (b) Variant 2 (c) Variant 3



(d) Variant 4 (e) Variant 5 (f) Variant 6



(g) Frequency representation for 32 variants generated from the same $\mathbf{x}$. Architecture : `Conformer`

Figure A.2: Multi-modal generation for the seismic data. The input $\mathbf{x}$, the high-frequency encoding aspect is replaced by a Gaussian distribution, $\mathcal{N}(0, I)$. The generated output should produce an infinity values likelihood to the targeted broadband data. $G_y(\mathrm{cat}(F_{xy}(\mathbf{x}), \mathcal{N}(0, I)) \sim \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_\infty\}$.

```
        self.opt = opt
```

```
        self.src = opt.dataroot
        self.features = self.get_features()
        self.meta_data_magnitude= None
        self.meta_data_depth    = None
        self.meta_data_pga      = None
        self.broadband_dataset  = None
        self.lowpass_dataset    = None
        self.highpass_dataset   = None
        self.filter_coef        = None
        self.workers = int(0.9*multiprocessing.cpu_count()-1)
        app.logger.info(f"number of workers :{self.workers+1}")
        app.logger.info(f"number of 3D-signals :{self.opt.nsy}")

    def replace_file_extension(self):
        *directories, last = self.src.split('/')
        last = last.replace('waveforms', 'metadata').replace('hdf5', 'csv')
        new_path = '/'.join(directories + [last])
        return new_path

    def parse_h5py_files(self):
        new_path = self.replace_file_extension()
        eqm = pd.read_csv(new_path)
        eqm = eqm.loc[eqm['trace_category'] == 'earthquake_local']
        eqm = eqm.loc[eqm['source_magnitude'] >= 3.5]
        eqm = eqm.sample(frac=self.opt.nsy/len(eqm)).reset_index(drop=True)
        w = windows.tukey(self.features['ntm'],5/100)

        return eqm, w

    def get_dataframe_from_stead(self, eqm):
        # Open the H5 file containing the waveform data
        with h5py.File(self.src, 'r') as f:
            # Create an empty DataFrame to store the waveform data
            dfs = []

            # Iterate through the rows of the eqm DataFrame
            for i, row in eqm.iterrows():
                # Get the trace name and p_arrival_sample value for this row
                tn = row['trace_name']
                bi = int(row['p_arrival_sample'])

                # Extract the waveform data for this trace name from the eqd H5 file
                waveform = f['earthquake']['local'][tn][:]

                # Create a new DataFrame with the waveform data and the p_arrival_sample value
                df_row = pd.DataFrame(waveform, columns=['{}_{}'.format(tn, j) for j in range(waveform.shape[1])])
                df_row['trace_name'] = tn
                df_row['p_arrival_sample'] = bi

                # Append the new DataFrame row to the main DataFrame
                dfs.append(df_row)
            df = pd.concat(dfs, ignore_index=True)
            return df


```

```python
    def get_features(self):
        md = {'dtm':0.01,'cutoff':self.opt.cutoff,'ntm':self.opt.imageSize}
        md['vTn'] = np.arange(0.0,3.05,0.05,dtype=np.float64)
        md['nTn'] = md['vTn'].size
        return md

    def process_broadband_data(self, args):
        start_idx, end_idx, src, opt, eqm, w = args
        with h5py.File(src, "r") as eqd:
            results = []
            _eqd = eqd['earthquake']['local']

            for i in tqdm(range(start_idx, end_idx), desc="sub processing datum", position=1, leave=False):
                tn = eqm.loc[i,'trace_name']
                magnitude = eqm.loc[i,'source_magnitude']
                depth = eqm.loc[i,'source_depth_km']
                bi = int(_eqd[tn].attrs['p_arrival_sample'])
                dataset  = np.zeros((3,opt.imageSize))
                lowpass_dataset = np.zeros((3,opt.imageSize))
                pgat_set = np.zeros((3))
                for j in range(3):
                    data = detrend(_eqd[tn][bi:bi+opt.imageSize, j]) * w
                    pgat = np.abs(data).max()
                    dataset_view = dataset[j, :].view()
                    lowpass_dataset_view = lowpass_dataset[j, :].view()
                    dataset_view[:] = data / pgat
                    lowpass_dataset_view[:] = self.filter_data(dataset_view[:], type='low')
                    pgat_set[j] = pgat
                results.append((dataset, lowpass_dataset, pgat_set, magnitude, depth))
            return results

    def process_extractions(self, eqm, w):
        with concurrent.futures.ProcessPoolExecutor(max_workers=self.workers) as executor:
            futures = []
            chunk_size = len(eqm.index) // self.workers
            chunks = [(i*chunk_size, (i+1)*chunk_size, self.src, self.opt, eqm, w) for i in range(self.workers-1)]
            chunks.append(((self.workers-1)*chunk_size, len(eqm.index), self.src, self.opt, eqm, w))
            for chunk in tqdm(chunks, total=len(chunks), desc="processing broadband", position=0):
                futures.append(executor.submit(self.process_broadband_data, chunk))

            results = []
            completed = set()
            remaining = set(futures)
            pbar = tqdm(total=len(futures), desc='collecting broadband')
            while remaining:
                done, remaining = concurrent.futures.wait(remaining,\
                    return_when=concurrent.futures.FIRST_COMPLETED)
                completed.update(done)
                for future in done:
                    content = future.result()
```

```
                 results.append(content)
                 pbar.update(1)
         pbar.close()
         data_list, pgat_list = [], []
         lowpass_list = []
         magnitude_list, depth_list  = [], []

         for i, content in enumerate(results):
             for data, low, pgat, magnitude, depth in content:
                 data_list.append(data)
                 lowpass_list.append(low)
                 pgat_list.append(pgat)
                 magnitude_list.append(magnitude)
                 depth_list.append(float(depth))

         dataset = torch.stack([torch.from_numpy(data) for data in data_list])
         pgat_set= torch.stack([torch.from_numpy(pgat) for pgat in pgat_list])
         lowpass_set = torch.stack([torch.from_numpy(low) for low in lowpass_list])

         magnitude_set =  torch.tensor(magnitude_list)
         depth_set = torch.tensor(depth_list)
         executor.shutdown()
         self.broadband_dataset = dataset
         self.meta_data_pga     = pgat_set
         self.meta_data_depth   = depth_set
         self.meta_data_magnitude = magnitude_set
         self.lowpass_dataset     = lowpass_set

 def filter_data(self, data, type='low'):
     # Apply lowpass filter to each channel of data using pre-built filter coefficients
         # If filter coefficients have not been initialized, initialize them
     nyquist_freq = 0.5 / self.features['dtm']
     cutoff_freq  = self.features['cutoff'] / nyquist_freq
     filter_coef = signal.butter(4, cutoff_freq, btype=type, output='sos')

     filtered_data= signal.sosfilt(filter_coef, data)

     # Additional processing steps here...
     return filtered_data

 def low_pass_filter_biquad(self, data):
     return  lowpass_biquad(data,1./self.features['dtm'],self.features['cutoff'])

 def process_filtered_data(self, data_chunk):
     return (self.filter_data(data_chunk,type='low'), self.filter_data(data_chunk, type='high'))

 def save_broadband_dataset(self):
     if not os.path.exists(self.opt.outf+"temporary"):
         os.makedirs(self.opt.outf+"temporary/")
     app.logger.info("Saving the broadband temporary file ...")
```

```
        torch.save(self.broadband_dataset, self.opt.outf+"temporary/broadband.pt")


    def save_filtered_dataset(self):
        if not os.path.exists(self.opt.outf+"temporary"):
            os.makedirs(self.opt.outf+"temporary/")
        app.logger.info("Saving the filtered temporary file ...")
        torch.save(self.lowpass_dataset, self.opt.outf+"temporary/lowpass.pt")
        torch.save(self.highpass_dataset, self.opt.outf+"temporary/highpass.pt")


    def save_meta_data(self):
        if not os.path.exists(self.opt.outf+"temporary"):
            os.makedirs(self.opt.outf+"temporary/")
        app.logger.info("Saving the eqm file ...")
        torch.save(self.meta_data_magnitude, self.opt.outf+"temporary/meta_data_magnitude.pt")
        torch.save(self.meta_data_depth, self.opt.outf+"temporary/meta_data_depth.pt")
        torch.save(self.meta_data_pga, self.opt.outf+"temporary/pga_broadband.pt")


    def extract(self):
        eqm, w = self.parse_h5py_files()
        self.process_extractions(eqm, w)
        self.save_broadband_dataset()
        self.save_meta_data()
        self.save_filtered_dataset()
```

Listing A.1: Python code for the extraction of the STEAD database

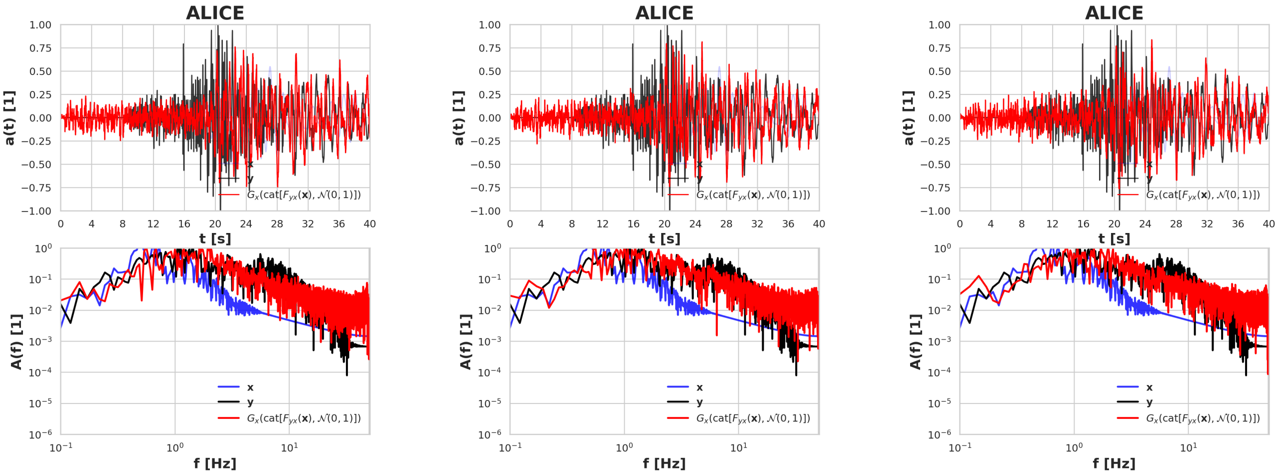
A.3.2 STEAD Loader

```
import torch

class STEADSetterDatateset(torch.utils.data.Dataset):
    def __init__(self, opt):
        super(STEADSetterDatateset, self).__init__()
        torch.manual_seed(100)
        self.opt = opt
        self.broadband_dataset          = torch.load(self.opt.dataroot+"temporary/broadband.pt")
        self.lowpass_filter_dataset     = torch.load(self.opt.dataroot+"temporary/lowpass.pt")
        self.meta_data_depth            = torch.load(self.opt.dataroot+"temporary/meta_data_depth.pt")
        self.meta_data_magnitude        = torch.load(self.opt.dataroot+"temporary/meta_data_magnitude.pt")
        self.pga_broadband              = torch.load(self.opt.dataroot+"temporary/pga_broadband.pt")

    def __len__(self):
        return len(self.broadband_dataset)

    def __getitem__(self,index):
```

```python
        broadband   = self.broadband_dataset[index,:,:].float()
        lowpass     = self.lowpass_filter_dataset[index,:,:].float()
        depth       = self.meta_data_depth[index].float()
        magnitude   = self.meta_data_magnitude[index].float()
        pga         = self.pga_broadband[index,:].float()

        return broadband, lowpass, pga, depth, magnitude


def get_dataset(opt, dataset=STEADSetterDatateset, batch_size=1, rank = 0, world_size = 1):
    _dataset = dataset(opt)

    train_part, vld_part = int(0.80*len(_dataset)), int(0.10*len(_dataset))
    tst_part = len(_dataset) - train_part - vld_part

    train_set, vld_set, tst_set   = torch.utils.data.random_split(_dataset, [train_part,vld_part,tst_part])

    # define DataLoader
    batch_size = opt.batchSize                # adjust batch size according to GPU type (16GB or 32GB in memory)
    drop_last = True                          # set to False if it represents important information loss
    num_workers = opt.workers                 # adjust number of CPU workers per process
    persistent_workers = True                 # set to False if CPU RAM must be released
    pin_memory = True                         # optimize CPU to GPU transfers
    prefetch_factor = 2                       # adjust number of batches to preload
    shuffle = True                            # set to True to have the data reshuffled at every epoch

    trn_loader = torch.utils.data.DataLoader(dataset=train_set,
            batch_size=batch_size,
            drop_last=drop_last,
            num_workers=num_workers,
            persistent_workers=persistent_workers,
            pin_memory=pin_memory,
            prefetch_factor=prefetch_factor,
            shuffle=shuffle,
        )

    vld_loader = torch.utils.data.DataLoader(dataset=vld_set,
            batch_size=batch_size,
            drop_last=drop_last,
            num_workers=num_workers,
            persistent_workers=persistent_workers,
            pin_memory=pin_memory,
            prefetch_factor=prefetch_factor,
            shuffle=shuffle,
        )

    tst_loader = torch.utils.data.DataLoader(dataset=tst_set,
            batch_size=batch_size,
            drop_last=drop_last,
            num_workers=num_workers,
```

```
            persistent_workers=persistent_workers,
            pin_memory=pin_memory,
            prefetch_factor=prefetch_factor,
            shuffle=shuffle,
        )

    return trn_loader, vld_loader, tst_loader
```

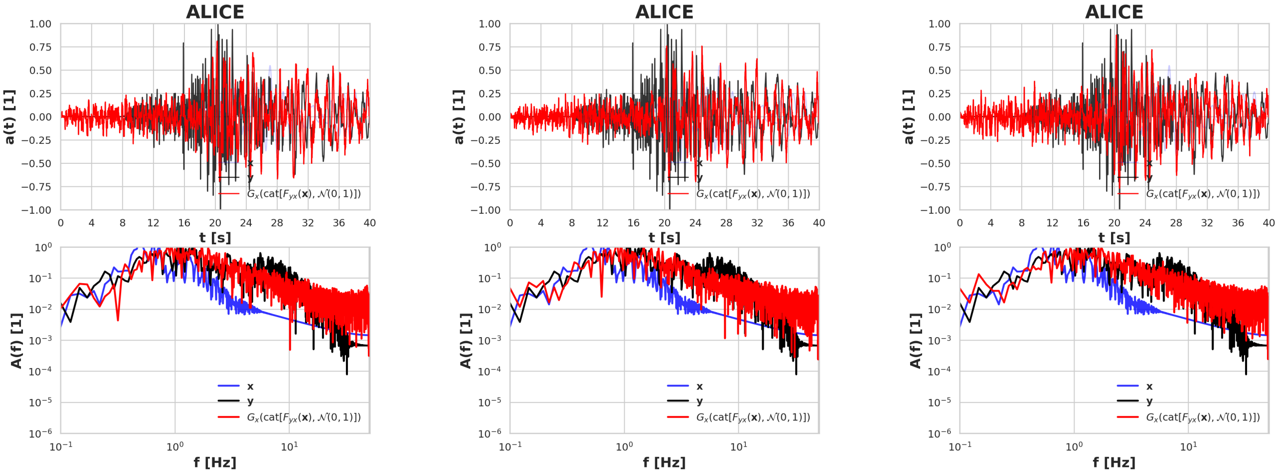
Listing A.2: Python code to load the STEAD database



(a) Variant 1 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$

(b) Variant 2 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$

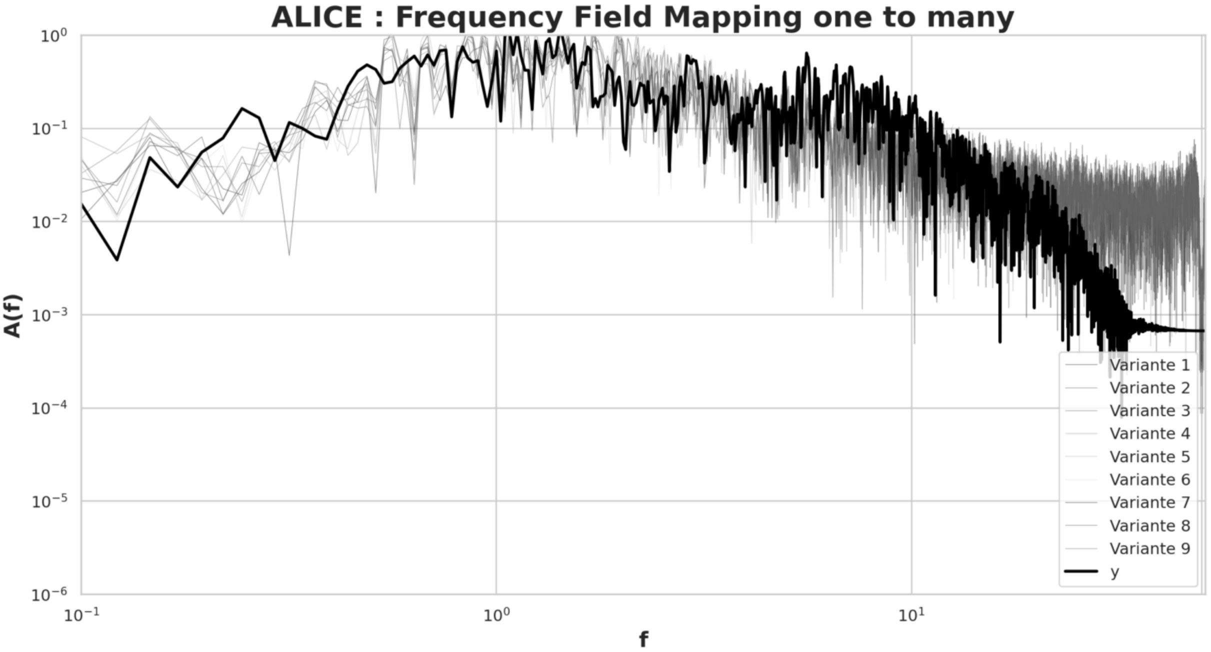
(c) Variant 3 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$



(d) Variant 4 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$

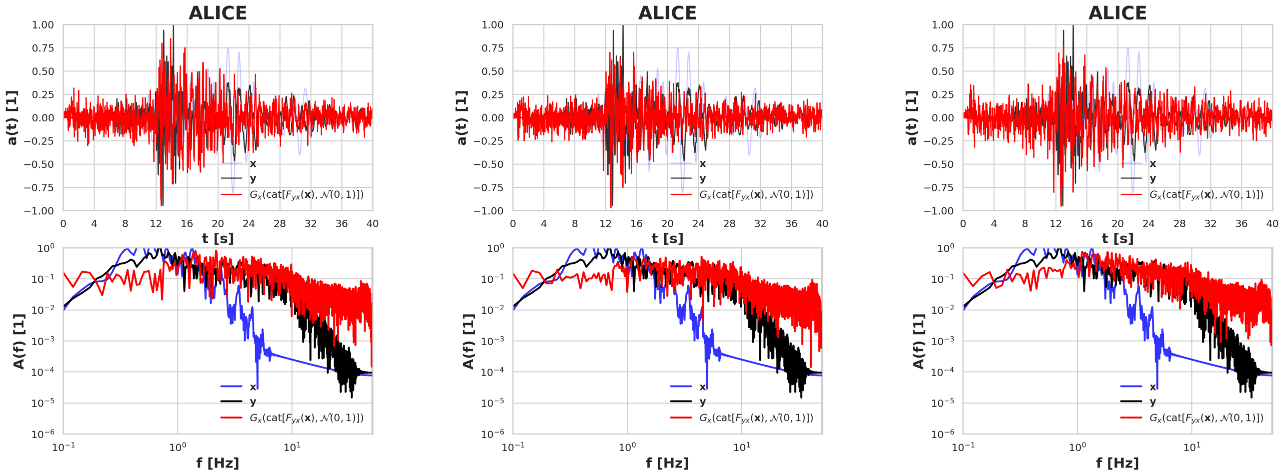
(e) Variant 5 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$

(f) Variant 6 of $G_y(F_{xy}^{xy}(\mathbf{x}), \mathcal{N}(0,I))$



(g) Frequency Filed of the different variants from Splitted latent space of `ALICE`

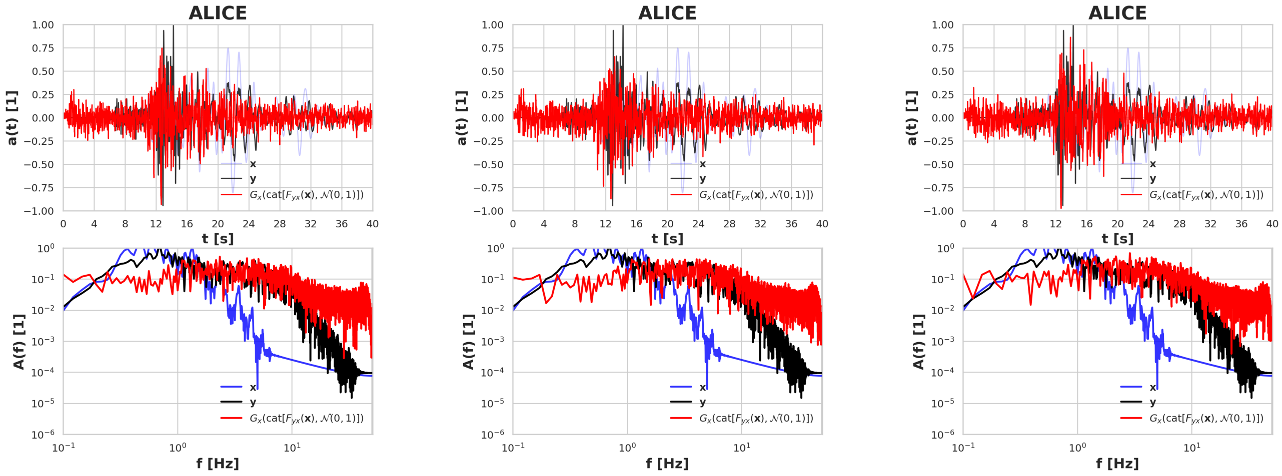
Figure A.3: SeismoALICE with splitted latent space. Seismogram OGCB is used. We have presented the multiple variant for direction UD, as example



(a) Variant 1 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$

(b) Variant 2 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$

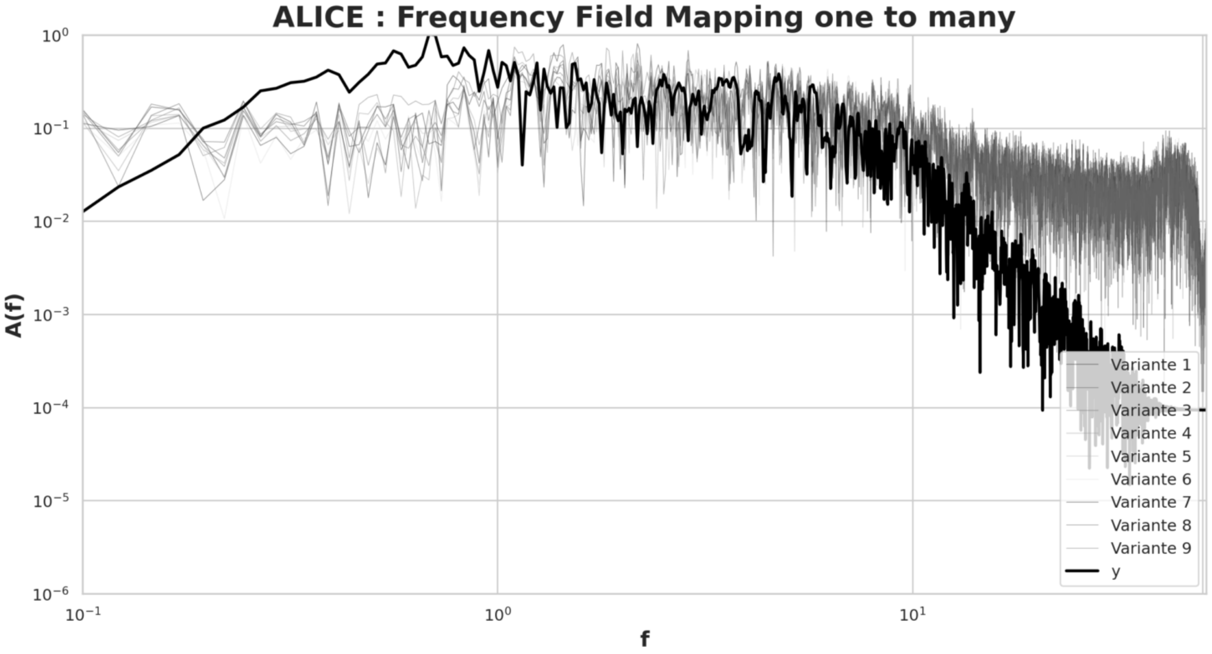
(c) Variant 3 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$



(d) Variant 4 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$

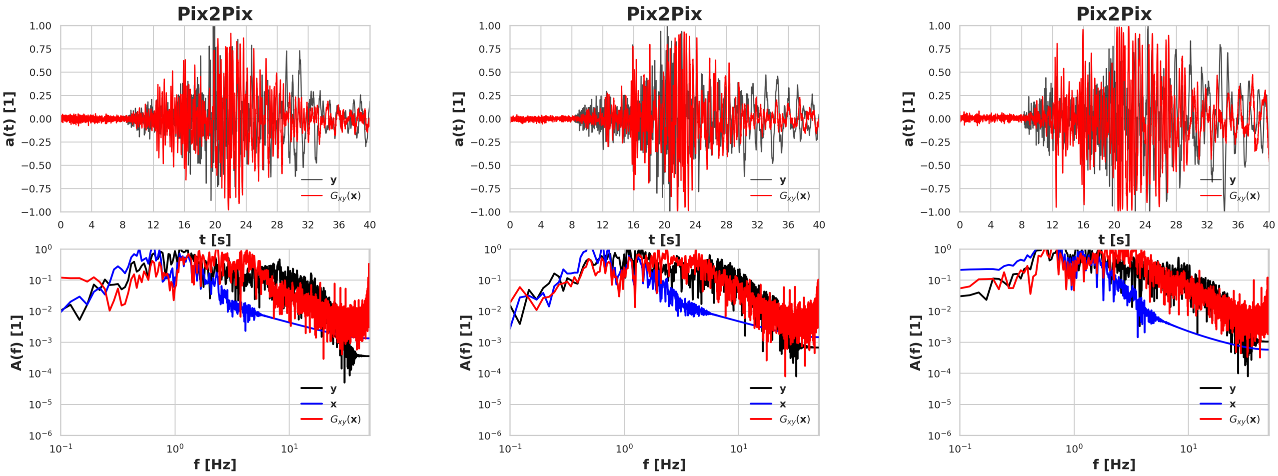
(e) Variant 5 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$

(f) Variant 6 of $G_y(F_{xy}^{xy}(\mathbf{x}),\mathcal{N}(0,I))$



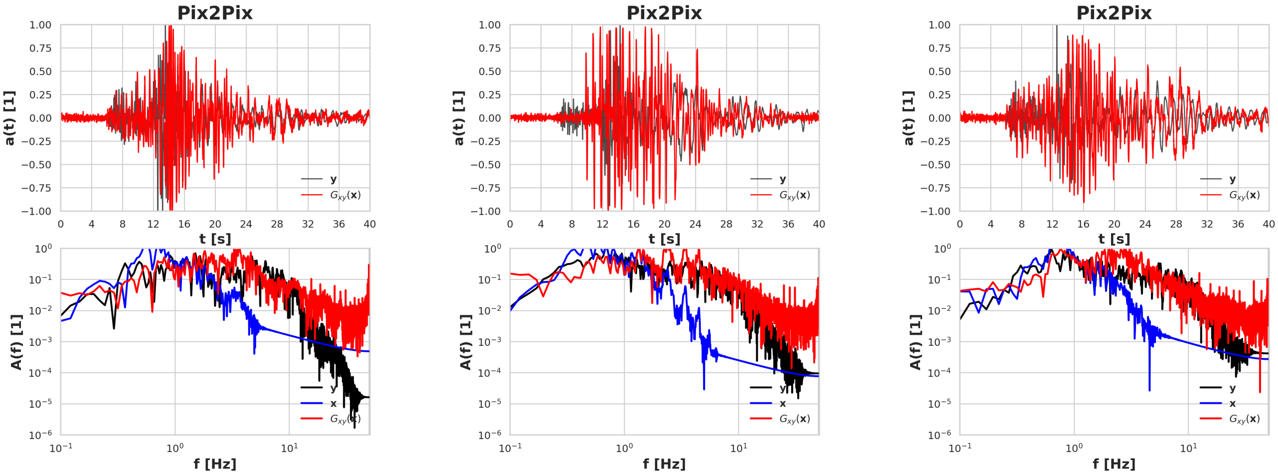
(g) Frequency Filed of the different variants from Splitted latent space of `ALICE`

Figure A.4: SeismoALICE with splitted latent space. Seismogram OGDF is used. We have presented the multiple variant for direction UD, as example



(a) NS (b) EW (c) UD

Figure A.5: Result for for the seismogram OCB for the simulation of Teil



(a) NS (b) EW (c) UD

Figure A.6: Results for for the seismogram OGDF for the simulation of Teil. The generation is poor because there is a factor of 10 difference between the PBS amplitude and that of the broadband signal.

Bibliography

[1] Ewald Brückl and Christa Hammerl. Eduard suess' conception of the alpine orogeny related to geophysical data and models.

[2] Joe J. Litehiser. *Observatory Seismology: An Anniversary Symposium on the Occasion of the Centennial of the University of California at Berkeley Seismographic Stations.* University of California Press. Google-Books-ID: yLJWcm0HXNIC.

[3] R. M. W. Musson. A history of british seismology. 11(3):715–861.

[4] Sheng Wang and Hong Hao. Effects of random variations of soil properties on site amplification of seismic ground motions. 22(7):551–564.

[5] D. Breysse, H. Niandou, S. Elachachi, and L. Houy. A generic approach to soil–structure interaction considering the effects of soil heterogeneity. 55(2):143–150.

[6] Edith Sotelo, Marco Favino, and Richard L. Gibson Jr. Application of the generalized finite-element method to the acoustic wave simulation in exploration seismology. 86(1):T61–T74.

[7] Marco Stupazzini, Maria Infantino, Alexander Allmann, and Roberto Paolucci. Physics-based probabilistic seismic hazard and loss assessment in large urban areas: A simplified application to istanbul. 50(1):99–115. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.3365.

[8] Samaneh Dashti, Karim Tarbali, Cong Zhou, and James Geoffrey Chase. Probabilistic seismic performance assessment of offshore jacket structures as a basis for engineering decision-making. 0(0):1–13. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15732479.2022.2132519.

[9] Marco Stupazzini, Maria Infantino, Alexander Allmann, and Roberto Paolucci. Physics-based probabilistic seismic hazard and loss assessment in large urban areas: A simplified application to istanbul. 50(1):99–115. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eqe.3365.

[10] Huang J, Motha J, Tarbali K, Lee R, Bae S, Polak V, Zhu M, Schill C, Paterson J, Lagrava D, and Brendon Bradley. Cybershake NZ v19.5: New zealand simulation-based probabilistic seismic hazard analysis.

[11] Nobuhito Mori, Tomoyuki Takahashi, Tomohiro Yasuda, and Hideaki Yanagisawa. Survey of 2011 tohoku earthquake tsunami inundation and run-up. 38(7). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2011GL049210.

[12] H Kit Miyamoto and Tom Chan. Assessment of damage and lessons learned.

[13] Earthquake damage in türkiye estimated to exceed $34 billion: World bank disaster assessment report.

[14] Cvetan Sinadinovski. The event of 26th of december 2004 – the biggest earthquake in the world in the last 40 years. 4(2):131–139.

[15] 2010 haiti earthquake | magnitude, damage, map, & facts | britannica.

[16] F. Gatti, S. Touhami, F. Lopez-Caballero, R. Paolucci, D. Clouteau, V. Alves Fernandes, M. Kham, and F. Voldoire. Broad-band 3-d earthquake simulation at nuclear site by an all-embracing source-to-structure approach. 115:263–280.

[17] Mohamed Sallak, Felipe Aguirre, and Walter Schon. Incertitudes aléatoires et épistémiques, comment les distinguer et les manipuler dans les études de fiabilité? page 9.

[18] S. Mostafa Mousavi and Gregory C. Beroza. Machine learning in earthquake seismology. 51(1):105–129. _eprint: https://doi.org/10.1146/annurev-earth-071822-100323.

[19] Giuseppe Carlo Marano, Marco Martino Rosso, Angelo Aloisio, and Giansalvo Cirrincione. Generative adversarial networks review in earthquake-related engineering fields.

[20] Victor Davidovici. *Conception-construction parasismique*. Collection Eurocode. Eyrolles Afnor éditions.

[21] Filippo Gatti and Didier Clouteau. Towards blending physics-based numerical simulations and seismic databases using generative adversarial network. 372:113421.

[22] Kevin P. Murphy. *Probabilistic machine learning: an introduction*. Adaptive computation and machine learning series. The MIT Press.

[23] S. R. McNutt. *Seismic Monitoring and Eruption Forecasting of Volcanoes: A Review of the State-of-the-Art and Case Histories*, pages 99–146. Springer Berlin Heidelberg.

[24] Amirul Sadikin Md Affendi and Marina Yusoff. Review of anomalous sound event detection approaches. 8(3):264.

[25] Patrick Laumann, Nishtha Srivastava, Wei Li, and Georg Ruempker. Volcano-seismic event classification using wavelet scattering transforms. Conference Name: EGU23.

[26] Seismic discrimination with artificial neural networks: Preliminary results with regional spectral data | bulletin of the seismological society of america | GeoScienceWorld.

[27] Lisa Linville, Kristine Pankow, and Timothy Draelos. Deep learning models augment analyst decisions for event discrimination. 46(7):3643–3651.

[28] Rigobert Tibi, Lisa Linville, Christopher Young, and Ronald Brogan. Classification of local seismic events in the utah region: A comparison of amplitude ratio methods with a spectrogram-based machine learning approach. 109(6):2532–2544.

[29] Haitham M. Dawood and Adrian Rodriguez-Marek. A method for including path effects in ground-motion prediction equations: An example using the mw 9.0 tohoku earthquake aftershocks. 103(2):1360–1372.

[30] Angel Bueno Rodriguez, Carmen Benitez, Luciano Zuccarello, Silvio De Angelis, and Jesus M. Ibanez. Bayesian monitoring of seismo-volcanic dynamics. 60:1–14.

[31] M. Masotti, S. Falsaperla, H. Langer, S. Spampinato, and R. Campanini. Application of support vector machine to the classification of volcanic tremor at etna, italy. 33(20):L20304.

[32] S. Mostafa Mousavi and Gregory C. Beroza. A machine-learning approach for earthquake magnitude estimation. 47(1).

[33] Pingan Peng, Zhengxiang He, Liguan Wang, and Yuanjian Jiang. Microseismic records classification using capsule network with limited training samples in underground mining. 10(1):13925. Number: 1 Publisher: Nature Publishing Group.

[34] Malcolm C. A. White, Kushal Sharma, Ang Li, T. K. Satish Kumar, and Nori Nakata. Classifying seismograms using the FastMap algorithm and support-vector machines. 2(1):1–10. Number: 1 Publisher: Nature Publishing Group.

[35] Guilherme Madureira, António E. Ruano, and Maria Graça Ruano. On-line operation of an intelligent seismic detector. In Valentina Emilia Balas, János Fodor, Annamária R. Várkonyi-Kóczy, Joszef Dombi, and Lakhmi C. Jain, editors, *Soft Computing Applications*, Advances in Intelligent Systems and Computing, pages 531–542. Springer.

[36] Moritz Beyreuther and Joachim Wassermann. Continuous earthquake detection and classification using discrete hidden markov models. 175(3):1055–1066.

[37] S. Mostafa Mousavi, Weiqiang Zhu, Yixiao Sheng, and Gregory C. Beroza. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. 9(1):10267. Number: 1 Publisher: Nature Publishing Group.

[38] Gwantae Kim, Bonhwa Ku, and Hanseok Ko. Multifeature fusion-based earthquake event classification using transfer learning. 18(6):974–978.

[39] Omar M. Saad and Yangkang Chen. CapsPhase: Capsule neural network for seismic phase classification and picking. 60:1–11.

[40] S. Mostafa Mousavi, William L. Ellsworth, Weiqiang Zhu, Lindsay Y. Chuang, and Gregory C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. 11(1):3952. Number: 1 Publisher: Nature Publishing Group.

[41] Fernando Lara, Román Lara-Cueva, Julio C. Larco, Enrique V. Carrera, and Rubén León. A deep learning approach for automatic recognition of seismo-volcanic events at the cotopaxi volcano. 409:107142.

[42] Kaiwen Wang, William L. Ellsworth, Gregory C. Beroza, Gordon Williams, Miao Zhang, Dustin Schroeder, and Justin Rubinstein. Seismology with dark data: Image-based processing of analog records using machine learning for the rangely earthquake control experiment. 90(2):553–562.

[43] Pablo D. Hernandez, Jaime A. Ramirez, and Marcelo A. Soto. Deep-learning-based earthquake detection for fiber-optic distributed acoustic sensing. 40(8):2639–2650.

[44] Amanda M. Thomas, Asaf Inbal, Jacob Searcy, David R. Shelly, and Roland Bürgmann. Identification of low-frequency earthquakes on the san andreas fault with deep learning. 48(13):e2021GL093157. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021GL093157.

[45] Claudia Hulbert, Bertrand Rouet-Leduc, Romain Jolivet, and Paul A. Johnson. An exponential build-up in seismic energy suggests a months-long nucleation of slow slip in cascadia. 11(1):4139. Number: 1 Publisher: Nature Publishing Group.

[46] S. Mostafa Mousavi, William L. Ellsworth, Weiqiang Zhu, Lindsay Y. Chuang, and Gregory C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. 11(1):3952. Number: 1 Publisher: Nature Publishing Group.

[47] David R. Boden. *Geology and Heat Architecture of the Earth's Interior.* Routledge Handbooks Online.

[48] Conny Hammer, Moritz Beyreuther, and Matthias Ohrnberger. A seismic-event spotting system for volcano fast-response systems. 102(3):948–960.

[49] Shaobo Yang, Jing Hu, Haijiang Zhang, and Guiquan Liu. Simultaneous earthquake detection on multiple stations via a convolutional neural network. 92(1):246–260.

[50] Sacha Lapins, Berhe Goitom, J-Michael Kendall, Maximilian J. Werner, Katharine V. Cashman, and James O. S. Hammond. A little data goes a long way: Automating seismic phase arrival picking at nabro volcano with transfer learning. 126(7):e2021JB021910.

[51] Jack Woollam, Jannes Münchmeyer, Frederik Tilmann, Andreas Rietbrock, Dietrich Lange, Thomas Bornstein, Tobias Diehl, Carlo Giunchi, Florian Haslinger, Dario Jozinović, Alberto Michelini, Joachim Saul, and Hugo Soto. SeisBench—a toolbox for machine learning in seismology. 93(3):1695–1709.

[52] Wei Li, Megha Chakraborty, Claudia Quinteros Cartaya, Jonas koehler, Johannes Faber, Georg Ruempker, and Nishtha Srivastava. SAIPy: A python package for single station earthquake monitoring using deep learning.

[53] Zachary E. Ross, Men-Andrin Meier, and Egill Hauksson. P wave arrival picking and first-motion polarity determination with deep learning. 123(6):5120–5129. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2017JB015251.

[54] Shota Hara, Yukitoshi Fukahata, and Yoshihisa Iio. P-wave first-motion polarity determination of waveform data in western japan using deep learning. 71(1):127.

[55] S. Mostafa Mousavi, Weiqiang Zhu, William Ellsworth, and Gregory Beroza. Unsupervised clustering of seismic signals using deep convolutional autoencoders. 16(11):1693–1697.

[56] Takahiko Uchide. Focal mechanisms of small earthquakes beneath the japanese islands based on first-motion polarities picked using deep learning. 223(3):1658–1671.

[57] Zachary E. Ross, Yisong Yue, Men-Andrin Meier, Egill Hauksson, and Thomas H. Heaton. PhaseLink: A deep learning approach to seismic phase association. 124(1):856–869. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018JB016674.

[58] Weiqiang Zhu, Ian W. McBrearty, S. Mostafa Mousavi, William L. Ellsworth, and Gregory C. Beroza. Earthquake phase association using a bayesian gaussian mixture model. 127(5):e2021JB023249.

[59] Joshua Dickey, Brett Borghetti, William Junek, and Richard Martin. Beyond correlation: A path-invariant measure for seismogram similarity. 91(1):356–369.

[60] Maren Böse, Friedemann Wenzel, and Mustafa Erdik. PreSEIS: A neural network-based approach to earthquake early warning for finite faults. 98(1):366–382.

[61] Probabilistic point source inversion of strong-motion data in 3-d media using pattern recognition: A case study for the 2008 mw 5.4 chino hills earthquake - käufl - 2016 - geophysical research letters - wiley online library.

[62] Anthony Lomax, Alberto Michelini, and Dario Jozinović. An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network. 90(2):517–529.

[63] S. Mostafa Mousavi and Gregory C. Beroza. Bayesian-deep-learning estimation of earthquake location from single-station observations. 58(11):8211–8224.

[64] Nicolae-Catalin Ristea and Anamaria Radoi. Complex neural networks for estimating epicentral distance, depth, and magnitude of seismic waves. 19:1–5.

[65] Andrea Licciardi, Quentin Bletery, Bertrand Rouet-Leduc, Jean-Paul Ampuero, and Kévin Juhel. Instantaneous tracking of earthquake growth with elastogravity signals. 606(7913):319–324. Number: 7913 Publisher: Nature Publishing Group.

[66] Marius Kriegerowski, Gesa M. Petersen, Hannes Vasyura-Bathke, and Matthias Ohrnberger. A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms. 90(2):510–516.

[67] Xiong Zhang, Jie Zhang, Congcong Yuan, Sen Liu, Zhibo Chen, and Weiping Li. Locating earthquakes with a network of seismic stations via a deep learning method. page 14.

[68] J.-T. Lin, D. Melgar, A. M. Thomas, and J. Searcy. Early warning for great earthquakes from characterization of crustal deformation patterns with deep learning. 126(10):e2021JB022703. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021JB022703.

[69] M. P. A. Van Den Ende and J.-P. Ampuero. Automated seismic source characterization using deep graph neural networks. 47(17):e2020GL088690.

[70] Jannes Münchmeyer, Dino Bindi, Ulf Leser, and Frederik Tilmann. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network. 226(2):1086–1104.

[71] Tai-Lin Chin, Kuan-Yu Chen, Da-Yi Chen, and Te-Hsiu Wang. An attention-based hypocenter estimator for earthquake localization. 60:1–10.

[72] Jonthan D Smith, Zachary E Ross, Kamyar Azizzadenesheli, and Jack B Muir. HypoSVI: Hypocentre inversion with stein variational inference and physics informed neural networks. 228(1):698–710.

[73] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. 378:686–707.

[74] Shaikhah Alkhadhr and Mohamed Almekkawy. Wave equation modeling via physics-informed neural networks: Models of soft and hard constraints for initial and boundary conditions. 23(5):2792.

[75] Wenhuan Kuang, Congcong Yuan, and Jie Zhang. Real-time determination of earthquake focal mechanism via deep learning. 12(1):1432. Number: 1 Publisher: Nature Publishing Group.

[76] Andreas Steinberg, Hannes Vasyura-Bathke, Peter Gaebler, Matthias Ohrnberger, and Lars Ceranna. Estimation of seismic moment tensors using variational inference machine learning. 126(10):e2021JB022685. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021JB022685.

[77] Hongliang Zhang, Kristopher A. Innanen, and David W. Eaton. Inversion for shear-tensile focal mechanisms using an unsupervised physics-guided neural network. 92(4):2282–2294.

[78] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Solving the wave equation with physics-informed deep learning.

[79] Chao Song, Tariq Alkhalifah, and Umair Bin Waheed. Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks. 225(2):846–859.

[80] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. version: 1.

[81] Tiantong Wang, Daniel Trugman, and Youzuo Lin. SeismoGen: Seismic waveform synthesis using generative adversarial networks. Number: arXiv:1911.03966.

[82] Aming Wu, Juyong Shin, Jae-Kwang Ahn, and Young-Woo Kwon. Augmenting seismic data using generative adversarial network for low-cost MEMS sensors. 9:167140–167153. Conference Name: IEEE Access.

[83] Manuel A. Florez, Michaelangelo Caporale, Pakpoom Buabthong, Zachary E. Ross, Domniki Asimaki, and Men-Andrin Meier. Data-driven synthesis of broadband earthquake ground motions using artificial intelligence. 112(4):1979–1996.

[84] Yuanming Li, Bonhwa Ku, Gwantae Kim, Jae-Kwang Ahn, and Hanseok Ko. Seismic signal synthesis by generative adversarial network with gated convolutional neural network structure. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 3857–3860. ISSN: 2153-7003.

[85] Roberto Paolucci, Filippo Gatti, Maria Infantino, Chiara Smerzini, Ali Güney Özcebe, and Marco Stupazzini. Broadband ground motions from 3d physics-based numerical simulations using artificial neural NetworksBroadband ground motions from 3d PBSs using ANNs. 108(3):1272–1286. Publisher: GeoScienceWorld.

[86] Fabio Sabetta and Antonio Pugliese. Estimation of response spectra and simulation of nonstationary earthquake ground motions. 86(2):337–352.

[87] Boumédiène Derras, Pierre Yves Bard, and Fabrice Cotton. Towards fully data driven ground-motion prediction models for europe. 12(1):495–516.

[88] Manuel A. Florez, Michaelangelo Caporale, Pakpoom Buabthong, Zachary E. Ross, Domniki Asimaki, and Men-Andrin Meier. Data-driven synthesis of broadband earthquake ground motions using artificial intelligence. 112(4):1979–1996.

[89] Jinjun Hu, Chaoyue Jin, Hui Zhang, Lei Hu, and Zhongwei Wang. Support vector regression for developing ground-motion models for arias intensity, cumulative absolute velocity, and significant duration for the kanto region, japan. 93(3):1619–1635.

[90] Kyle B. Withers, Morgan P. Moschetti, and Eric M. Thompson. A machine learning approach to developing ground motion models from simulated ground motions. 47(6):e2019GL086690. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2019GL086690.

[91] Ting-Yu Hsu and Chao-Wen Huang. Onsite early prediction of PGA using CNN with multi-scale and multi-domain p-waves as input. 9.

[92] Lim Eng Aik, Lam Chee Kiang, Zulkifley Bin Mohamed, and Tan Wei Hong. A review on the multivariate statistical methods for dimensional reduction studies. page 020003.

[93] PG Student, CSE, MIT, Aurangabad, Maharashtra, India, Rahul R.Chakre, and Dr. Radhakrishna Naik. Performance analysis of hybrid (supervised and unsupervised) method for multiclass data set. 16(4):93–99.

[94] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. 15(6):993–1000.

[95] Ker-Chau Li, Yve Aragon, Kerby Shedden, and C Thomas Agnan. Dimension reduction for multivariate response data. 98(461):99–109.

[96] Karl Pearson. LIII. *On lines and planes of closest fit to systems of points in space.* 2(11):559–572.

[97] T. I. Katsaounis. Analyzing multivariate data. 46.

[98] X. Wan, Dong Wang, P. Tse, Guanghua Xu, and Qing Zhang. A critical study of different dimensionality reduction methods for gear crack degradation assessment under different operating conditions. 78.

[99] K. Unglert, V. Radić, and A. M. Jellinek. Principal component analysis vs. self-organizing maps combined with hierarchical clustering for pattern recognition in volcano seismic spectra. 320:58–74.

[100] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. 10(1):5416.

[101] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-SNE. 9:2579–2605.

[102] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction.

[103] UMAP: Uniform manifold approximation and projection for dimension reduction — umap 0.5 documentation.

[104] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. TimeVAE: A variational auto-encoder for multivariate time series generation.

[105] Jinsung Yoon and Daniel Jarrett. Time-series generative adversarial networks. page 11.

[106] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs.

[107] F. Varadi, J. M. Pap, R. K. Ulrich, L. Bertello, and C. J. Henney. Searching for signal in noise by random-lag singular spectrum analysis. 526(2):1052–1061.

[108] V. Moskvina and K. M. Schmidt. Approximate projectors in singular spectrum analysis. 24(4):932–942.

[109] Wenxi Zhu. Robust singular spectrum analysis. In *2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, pages 1165–1170.

[110] Gustav Bredell, Kyriakos Flouris, Krishna Chaitanya, Ertunc Erdil, and Ender Konukoglu. Explicitly minimizing the blur error of variational autoencoders.

[111] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability.

[112] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. version: 7 type: article.

[113] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR. ISSN: 2640-3498.

[114] Qianwen Meng, Hangwei Qian, Yong Liu, Yonghui Xu, Zhiqi Shen, and Lizhen Cui. Unsupervised representation learning for time series: A review.

[115] Mahmoud A. Helal, Seif Eldawlatly, and Mohamed Taher. Using autoencoders for feature enhancement in motor imagery brain-computer interfaces. In *Biomedical Engineering*. ACTAPRESS.

[116] M. Casella, P. Dolce, M. Ponticorvo, and D. Marocco. Autoencoders as an alternative approach to principal component analysis for dimensionality reduction. an application on simulated data from psychometric models.

[117] Fanny Roche, T. Hueber, Samuel Limier, and Laurent Girin. Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models.

[118] Tiago S. Nazare, Rodrigo F. de Mello, and Moacir A. Ponti. Are pre-trained CNNs good feature extractors for anomaly detection in surveillance videos?

[119] Sherif Abdulatif, Ruizhe Cao, and Bin Yang. CMGAN: Conformer-based metric-GAN for monaural speech enhancement.

[120] Shengkui Zhao and Bin Ma. D2former: A fully complex dual-path dual-decoder conformer network using joint complex masking and complex spectral mapping for monaural speech enhancement.

[121] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition.

[122] Andrew P. Valentine and Jeannot Trampert. Data space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data: Autoencoder networks for waveform data. 189(2):1183–1202.

[123] Mingliang Liu and Dario Grana. Ensemble-based seismic history matching with data reparameterization using convolutional autoencoder. In *SEG Technical Program Expanded Abstracts 2018*, pages 3156–3160. Society of Exploration Geophysicists.

[124] Loïc Viens and Chris Van Houtte. Denoising ambient seismic field correlation functions with convolutional autoencoders. 220(3):1521–1535. Publisher: Oxford University Press.

[125] Feng Qian, Zhangbo Liu, Yan Wang, Songjie Liao, Shengli Pan, and Guangmin Hu. DTAE: Deep tensor autoencoder for 3-d seismic data interpolation. 60:1–19. Publisher: IEEE.

[126] Felix Leeb, Stefan Bauer, M. Besserve, and B. Scholkopf. Exploring the latent space of autoencoders with interventional assays.

[127] Jaehoon Cha and Jeyan Thiyagalingam. Disentangling autoencoders (DAE).

[128] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. Number: arXiv:1312.6114.

[129] MATTHIAS SEEGER. GAUSSIAN PROCESSES FOR MACHINE LEARNING. Publisher: World Scientific Publishing Company.

[130] Georges Sfeir, Filipe Rodrigues, and Maya Abou-Zeid. Gaussian process latent class choice models. 136:103552.

[131] Maja Rudolph, Stefan Kurz, and Barbara Rakitsch. Hybrid modeling design patterns.

[132] Mingliang Liu, Dario Grana, and Leandro Passos De Figueiredo. Uncertainty quantification in stochastic inversion with dimensionality reduction using variational autoencoder. 87(2):M43–M58.

[133] Chunxiao Ning and Yazhou Xie. Convolutional variational autoencoder for ground motion classification and generation toward efficient seismic fragility assessment. 39(2):165–185. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.13061.

[134] Haowei Hua, Feng Qian, Gulan Zhang, and Yuehua Yue. Unsupervised seismic facies deep clustering via lognormal mixture-based variational autoencoder. 16:9831–9842.

[135] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing.

[136] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks.

[137] Dan Li, Dacheng Chen, Lei Shi, Baihong Jin, Jonathan Goh, and See-Kiong Ng. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks.

[138] Kaleb E. Smith and Anthony O. Smith. Conditional GAN for timeseries generation.

[139] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs.

[140] Yuanming Li, Bonhwa Ku, Gwantae Kim, Jae-Kwang Ahn, and Hanseok Ko. Seismic signal synthesis by generative adversarial network with gated convolutional neural network structure. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 3857–3860. ISSN: 2153-7003.

[141] Zefeng Li, Men-Andrin Meier, Egill Hauksson, Zhongwen Zhan, and Jennifer Andrews. Machine learning seismic wave discrimination: Application to earthquake early warning. 45(10):4773–4779. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018GL077870.

[142] J. Kim Torbol, M. Generative adversarial network for earthquake early warning system. In *Earthquake Geotechnical Engineering for Protection and Development of Environment and Constructions*. CRC Press. Num Pages: 5.

[143] Men-Andrin Meier, Zachary E. Ross, Anshul Ramachandran, Ashwin Balakrishna, Suraj Nair, Peter Kundzicz, Zefeng Li, Jennifer Andrews, Egill Hauksson, and Yisong Yue. Reliable real-time seismic signal/noise discrimination with machine learning. 124(1):788–800. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2018JB016661.

[144] Heyi Liu, Shanyou Li, and Jindong Song. Discrimination between earthquake p waves and microtremors via a generative adversarial network. 112(2):669–679.

[145] Heyi Liu, Jindong Song, and Shanyou Li. Seismic event identification based on a generative adversarial network and support vector machine. 10.

[146] Jian Li, Dongwei Hei, Gaofeng Cui, Mengmin He, Juan Wang, Zhehan Liu, Jie Shang, Xiaoming Wang, and Weidong Wang. GAN-LSTM joint network applied to seismic array noise signal recognition. 11(21):9987. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.

[147] Tiantong Wang, Zhongping Zhang, and Youzuo Li. EarthquakeGen: Earthquake generator using generative adversarial networks. In *SEG Technical Program Expanded Abstracts 2019*, SEG Technical Program Expanded Abstracts, pages 2674–2678. Society of Exploration Geophysicists.

[148] Tiantong Wang, Daniel Trugman, and Youzuo Lin. SeismoGen: Seismic waveform synthesis using GAN with application to seismic data augmentation. 126(4):e2020JB020077. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2020JB020077.

[149] Yuanming Li, Bonhwa Ku, Shou Zhang, Jae-Kwang Ahn, and Hanseok Ko. Seismic data augmentation based on conditional generative adversarial networks. 20(23):6850. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[150] Yinjun Ding, Jun Chen, and Jiaxu Shen. Conditional generative adversarial network model for simulating intensity measures of aftershocks. 139:106281.

[151] Yinjun Ding, Jun Chen, and Jiaxu Shen. Prediction of spectral accelerations of aftershock ground motion with deep learning method. 150:106951.

[152] Mehrshad Matinfar, Naser Khaji, and Goodarz Ahmadi. Deep convolutional generative adversarial networks for the generation of numerous artificial spectrum-compatible earthquake accelerograms using a limited number of ground motion records. 38(2):225–240. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12852.

[153] Felipe Grijalva, Washington Ramos, Noel Perez, Diego Benitez, Roman Lara, and Mario Ruiz. ESeismic-GAN: A generative model for seismic events from cotopaxi volcano. 14:7111–7120.

[154] Manuel A. Florez, Michaelangelo Caporale, Pakpoom Buabthong, Zachary E. Ross, Domniki Asimaki, and Men-Andrin Meier. Data-driven accelerogram synthesis using deep generative models.

[155] Reza D. D. Esfahani, Fabrice Cotton, Matthias Ohrnberger, and Frank Scherbaum. TFC-GAN: Nonstationary ground-motion simulation in the time–frequency domain using conditional generative adversarial network (CGAN) and phase retrieval methods.

[156] Gao Fan, Jun Li, Hong Hao, and Yu Xin. Data driven structural dynamic response reconstruction using segment based generative adversarial networks. 234:111970.

[157] Tomoyuki Yamada, Noriyuki Takahashi, and Hiroyuki Chida. Automation technology of seismic damage investigation for timber houses using deep learning. 27(67):1578–1583. Publisher: Architectural Institute of Japan.

[158] Automated structural design of shear wall residential buildings using generative adversarial networks - ScienceDirect.

[159] Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks - lu - 2022 - earthquake engineering & structural dynamics - wiley online library.

[160] Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis. *Artificial Neural Networks and Machine Learning – ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part II.* Springer. Google-Books-ID: UGBwDwAAQBAJ.

[161] Takaya Ueda, Masataka Seo, and Ikuko Nishikawa. Data correction by a generative model with an encoder and its application to structure design. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, Lecture Notes in Computer Science, pages 403–413. Springer International Publishing.

[162] Sofia Tilon, Francesco Nex, Norman Kerle, and George Vosselman. Post-disaster building damage detection from earth observation imagery using unsupervised and transferable anomaly detecting generative adversarial networks. 12(24):4193. Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

[163] N. Zhang, F. Nex, G. Vosselman, and N. Kerle. UNSUPERVISED HARMONIOUS IMAGE COMPOSITION FOR DISASTER VICTIM DETECTION. XLIII-B3-2022:1189–1196. Conference Name: XXIV ISPRS Congress “Imaging today, foreseeing tomorrow”, Commission III - 2022 edition, 6–11 June 2022, Nice, France Publisher: Copernicus GmbH.

[164] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization.

[165] Minhyeok Lee and Junhee Seok. Regularization methods for generative adversarial networks: An overview of recent studies. page 18.

[166] Harsh Rangwani, Naman Jaswani, Tejan Karmali, Varun Jampani, and R. Venkatesh Babu. Improving GANs for long-tailed data through group spectral regularization.

[167] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks.

[168] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[169] Thomas Kurbiel and Shahrzad Khaleghian. Training of deep neural networks based on distance measures using RMSProp.

[170] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization.

[171] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. version: 6 type: article.

[172] Yuqian Zhou, Kuangxiao Gu, and Thomas Huang. Unsupervised representation adversarial learning network: from reconstruction to generation.

[173] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. Number: arXiv:1907.02544.

[174] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. version: 3.

[175] Chunyuan Li, Hao Liu, Changyou Chen, Yunchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. ALICE: Towards understanding adversarial learning for joint distribution matching.

[176] Junshu Tang, Zhiwen Shao, and Lizhuang Ma. Fine-grained expression manipulation via structured latent space. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[177] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3d guided fine-grained face manipulation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9813–9822. IEEE.

[178] Lorenzo Tronchin, Minh H. Vu, Paolo Soda, and Tommy Löfstedt. LatentAugment: Data augmentation via guided manipulation of GAN's latent space. Publisher: arXiv Version Number: 1.

[179] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. GAN-control: Explicitly controllable GANs.

[180] Firas Shama, Roey Mechrez, Alon Shoshan, and Lihi Zelnik-Manor. Adversarial feedback loop. pages 3205–3214.

[181] Ali Pourramezan Fard, Mohammad H. Mahoor, Sarah Ariel Lamer, and Timothy Sweeny. GANalyzer: Analysis and manipulation of GANs latent space for controllable face synthesis.

[182] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images.

[183] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning.

[184] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. StyleGAN2 distillation for feed-forward image manipulation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 170–186. Springer International Publishing.

[185] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. version: 2.

[186] SAHIL GOYAL. BicycleGAN.

[187] Benoît Oriol and Alexandre Miot. On some theoretical limitations of generative adversarial networks.

[188] Zhenyu Yang, Yantao Li, and Gang Zhou. TS-GAN: Time-series GAN for sensor-based health data augmentation. 4(2):12:1–12:21.

[189] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification.

[190] Jeff Z HaoChen, Adrien Gaidon, Colin Wei, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss.

[191] Xinyu Yang, Zhenguo Zhang, and Rongyi Cui. TimeCLR: A self-supervised contrastive learning framework for univariate time series representation. 245:108606.

[192] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. pages 9729–9738.

[193] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning.

[194] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding.

[195] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. pages 15750–15758.

[196] Kristoffer Wickstrøm, Michael Kampffmeyer, Karl Øyvind Mikalsen, and Robert Jenssen. Mixing up contrastive learning: Self-supervised representation learning for time series. 155:54–61.

[197] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations.

[198] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc.

[199] Vivek Sharma, Makarand Tapaswi, M. Saquib Sarfraz, and Rainer Stiefelhagen. Clustering based contrastive learning for improving face representations. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 109–116.

[200] Dejiao Zhang, Feng Nan, Xiaokai Wei, Shangwen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew Arnold, and Bing Xiang. Supporting clustering with contrastive learning.

[201] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. 35(10):8547–8555. Number: 10.

[202] Salar Hosseini Khorasgani, Yuxuan Chen, and Florian Shkurti. SLIC: Self-supervised learning with iterative clustering for human action videos. pages 16091–16101.

[203] MHCCL: Masked hierarchical cluster-wise contrastive learning for multivariate time series | proceedings of the AAAI conference on artificial intelligence.

[204] Zhixiong Yang, Junwen Pan, Yanzhan Yang, Xiaozhou Shi, Hong-Yu Zhou, Zhicheng Zhang, and Cheng Bian. ProCo: Prototype-aware contrastive learning for long-tailed medical image classification. Publisher: arXiv Version Number: 1.

[205] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2vec: Towards universal representation of time series. 36(8):8980–8987.

[206] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting.

[207] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding.

[208] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[209] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[210] Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *Proceedings of the 39th International Conference on Machine Learning*, pages 25038–25054. PMLR. ISSN: 2640-3498.

[211] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting.

[212] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 1d convolutional neural networks and applications: A survey. 151:107398.

[213] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis.

[214] Serkan Kiranyaz, Turker Ince, Ridha Hamila, and Moncef Gabbouj. Convolutional neural networks for patient-specific ECG classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2608–2611. IEEE.

[215] Yuma Kinoshita and Hitoshi Kiya. Checkerboard-artifact-free image-enhancement network considering local and global features.

[216] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.

[217] Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation. 27(8):1256–1266.

[218] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis.

[219] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need.

[220] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks.

[221] Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention.

[222] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. Publisher: arXiv Version Number: 3.

[223] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view.

[224] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. version: 3.

[225] Samir M. Perlaza, Gaetan Bisson, Iñaki Esnaola, Alain Jean-Marie, and Stefano Rini. Empirical risk minimization with generalized relative entropy regularization.

[226] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis.

[227] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec.

[228] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation.

[229] Shinji Takaki, Toru Nakashika, Xin Wang, and Junichi Yamagishi. STFT spectral loss for training a neural speech waveform model.

[230] Qiuyu Zhu, Hu Wang, and Ruixin Zhang. Wavelet loss function for auto-encoder. 9:27101–27108.

[231] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. MelGAN: Generative adversarial networks for conditional waveform synthesis.

[232] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. page 9.

[233] Miriam Kristeková, Jozef Kristek, Peter Moczo, and Steven M. Day. Misfit criteria for quantitative comparison of seismograms. 96(5):1836–1850.

[234] Miriam Kristeková, Jozef Kristek, and Peter Moczo. Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals. 178(2):813–825.

[235] S. Mostafa Mousavi, Yixiao Sheng, Weiqiang Zhu, and Gregory C. Beroza. STanford EArthquake dataset (STEAD): A global data set of seismic signals for AI. 7:179464–179476. Conference Name: IEEE Access.

[236] Dimitri Komatitsch, Gordon Erlebacher, Dominik Göddeke, and David Michéa. High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster. 229(20):7692–7714.

[237] Tsuyoshi Ichimura, Kohei Fujita, Kentaro Koyama, Yuma Kikuchi, Ryota Kusakabe, Kazuo Minami, Hikaru Inoue, Seiya Nishizawa, Miwako Tsuji, Tatsuo Nishiki, Muneo

Hori, Lalith Maddegedara, and Naonori Ueda. Fast scalable implicit solver with convergence of equation-based modeling and data-driven learning: earthquake city simulation on low-order unstructured finite element. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–12. ACM.

[238] David McCallen, Anders Petersson, Arthur Rodgers, Arben Pitarka, Mamun Miah, Floriana Petrone, Bjorn Sjogreen, Norman Abrahamson, and Houjun Tang. EQSIM—a multidisciplinary framework for fault-to-structure earthquake simulations on exascale computers part i: Computational models and workflow. 37(2):707–735.

[239] David McCallen, Floriana Petrone, Mamun Miah, Arben Pitarka, Arthur Rodgers, and Norman Abrahamson. EQSIM—a multidisciplinary framework for fault-to-structure earthquake simulations on exascale computers, part II: Regional simulations of building response. 37(2):736–761.

[240] Babak Poursartip, Arash Fathi, and John L. Tassoulas. Large-scale simulation of seismic wave motion: A review. 129:105909.

[241] Sara Touhami, Filippo Gatti, Fernando Lopez-Caballero, Régis Cottereau, Lúcio De Abreu Corrêa, Ludovic Aubry, and Didier Clouteau. SEM3d: A 3d high-fidelity numerical earthquake simulator for broadband (0–10 hz) seismic response prediction at a regional scale. 12(3):112.

[242] E. Chaljub, P. Moczo, S. Tsuno, P.-Y. Bard, J. Kristek, M. Kaser, M. Stupazzini, and M. Kristekova. Quantitative comparison of four numerical predictions of 3d ground motion in the grenoble valley, france. 100(4):1427–1455.

[243] Jacobo Bielak, Robert W. Graves, Kim B. Olsen, Ricardo Taborda, Leonardo RamÃrez GuzmÃ¡n, Steven M. Day, Geoffrey P. Ely, Daniel Roten, Thomas H. Jordan, Philip J. Maechling, John Urbanic, Yifeng Cui, and Gideon Juve. The ShakeOut earthquake scenario: Verification of three simulation sets. 180(1):375–404.

[244] Yifeng Cui, Kim B. Olsen, Thomas H. Jordan, Kwangyoon Lee, Jun Zhou, Patrick Small, Daniel Roten, Geoffrey Ely, Dhabaleswar K. Panda, Amit Chourasia, John Levesque, Steven M. Day, and Philip Maechling. Scalable earthquake simulation on petascale supercomputers. In *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–20. IEEE.

[245] Max Rietmann, Peter Messmer, Tarje Nissen-Meyer, Daniel Peter, Piero Basini, Dimitri Komatitsch, Olaf Schenk, Jeroen Tromp, Lapo Boschi, and Domenico Giardini. Forward and adjoint simulations of seismic wave propagation on emerging large-scale GPU architectures. In *2012 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE.

[246] Ricardo Taborda and Jacobo Bielak. Ground-motion simulation and validation of the 2008 chino hills, california, earthquake. 103(1):131–156.

[247] Alexander Heinecke, Alexander Breuer, Sebastian Rettenberger, Michael Bader, Alice-Agnes Gabriel, Christian Pelties, Arndt Bode, William Barth, Xiang-Ke Liao, Karthikeyan Vaidyanathan, Mikhail Smelyanskiy, and Pradeep Dubey. Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers. In

SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, pages 3–14. IEEE.

[248] Tsuyoshi Ichimura, Kohei Fujita, Seizo Tanaka, Muneo Hori, Maddegedara Lalith, Yoshihisa Shizawa, and Hiroshi Kobayashi. Physics-based urban earthquake simulation enhanced by 10.7 BlnDOF × 30 k time-step unstructured FE non-linear seismic wave simulation. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 15–26. IEEE.

[249] Haohuan Fu, Conghui He, Bingwei Chen, Zekun Yin, Zhenguo Zhang, Wenqiang Zhang, Tingjian Zhang, Wei Xue, Weiguo Liu, Wanwang Yin, Guangwen Yang, and Xiaofei Chen. 18.9-pflops nonlinear earthquake simulation on sunway TaihuLight: enabling depiction of 18-hz and 8-meter scenarios. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12. ACM.

[250] Kohei Fujita, Sota Murakami, Tsuyoshi Ichimura, Takane Hori, Muneo Hori, Maddegedara Lalith, and Naonori Ueda. Scalable finite-element viscoelastic crustal deformation analysis accelerated with data-driven method. In *2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH)*, pages 18–25. IEEE.

[251] Arnau Folch, Claudia Abril, Michael Afanasiev, Giorgio Amati, Michael Bader, Rosa M. Badia, Hafize B. Bayraktar, Sara Barsotti, Roberto Basili, Fabrizio Bernardi, Christian Boehm, Beatriz Brizuela, Federico Brogi, Eduardo Cabrera, Emanuele Casarotti, Manuel J. Castro, Matteo Cerminara, Antonella Cirella, Alexey Cheptsov, Javier Conejero, Antonio Costa, Marc De La Asunción, Josep De La Puente, Marco Djuric, Ravil Dorozhinskii, Gabriela Espinosa, Tomaso Esposti-Ongaro, Joan Farnós, Nathalie Favretto-Cristini, Andreas Fichtner, Alexandre Fournier, Alice-Agnes Gabriel, Jean-Matthieu Gallard, Steven J. Gibbons, Sylfest Glimsdal, José Manuel González-Vida, Jose Gracia, Rose Gregorio, Natalia Gutierrez, Benedikt Halldorsson, Okba Hamitou, Guillaume Houzeaux, Stephan Jaure, Mouloud Kessar, Lukas Krenz, Lion Krischer, Soline Laforet, Piero Lanucara, Bo Li, Maria Concetta Lorenzino, Stefano Lorito, Finn Løvholt, Giovanni Macedonio, Jorge Macías, Guillermo Marín, Beatriz Martínez Montesinos, Leonardo Mingari, Geneviève Moguilny, Vadim Montellier, Marisol Monterrubio-Velasco, Georges Emmanuel Moulard, Masaru Nagaso, Massimo Nazaria, Christoph Niethammer, Federica Pardini, Marta Pienkowska, Luca Pizzimenti, Natalia Poiata, Leonhard Rannabauer, Otilio Rojas, Juan Esteban Rodriguez, Fabrizio Romano, Oleksandr Rudyy, Vittorio Ruggiero, Philipp Samfass, Carlos Sánchez-Linares, Sabrina Sanchez, Laura Sandri, Antonio Scala, Nathanael Schaeffer, Joseph Schuchart, Jacopo Selva, Amadine Sergeant, Angela Stallone, Matteo Taroni, Solvi Thrastarson, Manuel Titos, Nadia Tonelllo, Roberto Tonini, Thomas Ulrich, Jean-Pierre Vilotte, Malte Vöge, Manuela Volpe, Sara Aniko Wirp, and Uwe Wössner. The EU center of excellence for exascale in solid earth (ChEESE): Implementation, results, and roadmap for the second phase. 146:47–61.

[252] Scott Callaghan, Philip Maechling, Karan Vahi, Ewa Deelman, Fabio Silva, Kevin Milner, Kim Olsen, Robert Graves, Thomas Jordan, and Yehuda Ben-Zion. Preparing seismic applications for exascale using scientific workflows.

[253] Samaneh Dashti, Karim Tarbali, Cong Zhou, and James Geoffrey Chase. Probabilistic seismic performance assessment of offshore jacket structures as a basis for

engineering decision-making. 0(0):1–13. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15732479.2022.2132519.

[254] Brendon A. Bradley, Didier Pettinga, Jack W. Baker, and Jeff Fraser. Guidance on the utilization of earthquake-induced ground motion simulations in engineering practice. 33(3):809–835.

[255] Kevin R. Milner, Bruce E. Shaw, Christine A. Goulet, Keith B. Richards-Dinger, Scott Callaghan, Thomas H. Jordan, James H. Dieterich, and Edward H. Field. Toward physics-based nonergodic PSHA: A prototype fully deterministic seismic hazard model for southern california. 111(2):898–915.

[256] Zihua Niu, Alice-Agnes Gabriel, and Heiner Igel. Numerical simulation and uncertainty quantification of models for coseismic damage and healing of rocks in 1d, 2d and 3d.

[257] David Castro-Cruz, Filippo Gatti, and Fernando Lopez-Caballero. High-fidelity broadband prediction of regional seismic response: a hybrid coupling of physics-based synthetic simulation and empirical green functions. 108(2):1997–2031.

[258] M. Korres, F. Lopez-Caballero, V. Alves Fernandes, F. Gatti, I. Zentner, F. Voldoire, D. Clouteau, and D. Castro-Cruz. Enhanced seismic response prediction of critical structures via 3d regional scale physics-based earthquake simulation. 27(3):546–574.

[259] F De Martin, E Chaljub, P Thierry, P Sochala, F Dupros, E Maufroy, B Hadri, A Benaichouche, and F Hollender. Influential parameters on 3-d synthetic ground motions in a sedimentary basin derived from global sensitivity analysis. 227(3):1795–1817.

[260] Roberto Paolucci, Ilario Mazzieri, Chiara Smerzini, and Marco Stupazzini. Physics-based earthquake ground shaking scenarios in large urban areas. In Atilla Ansal, editor, *Perspectives on European Earthquake Engineering and Seismology*, volume 34, pages 331–359. Springer International Publishing. Series Title: Geotechnical, Geological and Earthquake Engineering.

[261] Pierre Sochala, F. De Martin, and O. Le Maître. MODEL REDUCTION FOR LARGE-SCALE EARTHQUAKE SIMULATION IN AN UNCERTAIN 3d MEDIUM. 10(2):101–127.

[262] Arben Pitarka and Robert Mellors. Using dense array waveform correlations to build a velocity model with stochastic variability. 111(4):2021–2041.

[263] John M. Rekoske, Alice-Agnes Gabriel, and Dave A. May. Instantaneous physics-based ground motion maps using reduced-order modeling. 128(8):e2023JB026975.

[264] Fanny Lehmann, Filippo Gatti, Michaël Bertin, and Didier Clouteau. Fourier neural operator surrogate model to predict 3d seismic waves propagation.

[265] Katsuhiro Kamae, Kojiro Irikura, and Arben Pitarka. A technique for simulating strong ground motion using hybrid green's function. 88(2):357–367.

[266] C. Smerzini and M. Villani. Broadband numerical simulations in complex near-field geological configurations: The case of the 2009 mw 6.3 l'aquila earthquake. 102(6):2436–2451.

[267] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. Number: arXiv:1611.07004.

[268] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. E99.D(7):1877–1884.

[269] Younggun Lee and Taesu Kim. Robust and fine-grained prosody control of end-to-end speech synthesis.

[270] Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. Learning latent representations for style control and transfer in end-to-end speech synthesis.

[271] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks.

[272] Harpreet Kaur, Nam Pham, and Sergey Fomel. Seismic data interpolation using CycleGAN. In *SEG Technical Program Expanded Abstracts 2019*, pages 2202–2206. Society of Exploration Geophysicists.

[273] Takuhiro Kaneko and Hirokazu Kameoka. Parallel-data-free voice conversion using cycle-consistent adversarial networks.

[274] Wei Cai, Xiaoguang Li, and Lizuo Liu. A phase shift deep neural network for high frequency approximation and wave problems. 42(5):A3285–A3312. Publisher: Society for Industrial and Applied Mathematics.

[275] Ao Cai, Hongrui Qiu, and Fenglin Niu. Semi-supervised surface wave tomography with wasserstein cycle-consistent GAN: Method and application to southern california plate boundary region. 127(3):e2021JB023598. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021JB023598.

[276] Dowan Kim and Joongmoo Byun. Data augmentation using CycleGAN for overcoming the imbalance problem in petrophysical facies classification. In *SEG Technical Program Expanded Abstracts 2020*, pages 2310–2314. Society of Exploration Geophysicists.

[277] Zhi Zhong, Alexander Y. Sun, and Xinming Wu. Inversion of time-lapse seismic reservoir monitoring data using CycleGAN: A deep learning-based approach for estimating dynamic reservoir property changes. 125(3):e2019JB018408. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2019JB018408.

[278] Wenda Li and Jian Wang. Residual learning of cycle-GAN for seismic data denoising. 9:11585–11597.

[279] Francesco Picetti, Vincenzo Lipari, Paolo Bestagini, and Stefano Tubaro. A generative-adversarial network for seismic-imaging applications. In *SEG Technical Program Expanded Abstracts 2018*, SEG Technical Program Expanded Abstracts, pages 2231–2235. Society of Exploration Geophysicists.

[280] Wen Pan, Carlos Torres-Verdín, and Michael J. Pyrcz. Stochastic pix2pix: A new machine learning method for geophysical and well conditioning of rule-based channel reservoir models. 30(2):1319–1345.

[281] SangYeon Kim, Hyunwoo Lee, Jonghee Han, and Joon-Ho Kim. Sig2sig: Signal translation networks to take the remains of the past. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3620–3624. ISSN: 2379-190X.

[282] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. SEGAN: Speech enhancement generative adversarial network.

[283] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised dual learning for image-to-image translation.

[284] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks.

[285] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN.

[286] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks.

[287] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin.

[288] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery.

[289] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. version: 2.

[290] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs.

[291] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks.

[292] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. version: 7.

[293] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets.

[294] Yuanming Li, Bonhwa Ku, Shou Zhang, Jae-Kwang Ahn, and Hanseok Ko. Seismic data augmentation based on conditional generative adversarial networks. 20(23):6850. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[295] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors.

[296] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. Number: arXiv:1607.08022.

[297] Yuxin Wu and Kaiming He. Group normalization.

[298] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal PatchGAN. pages 9066–9075.

[299] Vincent Dumont, Xiangyang Ju, and Juliane Mueller. Hyperparameter optimization of generative adversarial network models for high-energy physics simulations.

[300] Papers with code - TTUR explained.

[301] Xiaomeng Dong, Tao Tan, Michael Potter, Yun-Chan Tsai, Gaurav Kumar, V. Ratna Saripalli, and Theodore Trafalis. Autonomous learning rate optimization for deep learning. In Dimitris E. Simos, Varvara A. Rasskazova, Francesco Archetti, Ilias S. Kotsireas, and Panos M. Pardalos, editors, *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 292–305. Springer International Publishing.

[302] Roberto Paolucci, Filippo Gatti, Maria Infantino, Chiara Smerzini, Ali Güney Özcebe, and Marco Stupazzini. Broadband ground motions from 3d physics-based numerical simulations using artificial neural networks. 108(3):1272–1286.

[303] Roberto Paolucci, Chiara Smerzini, and Manuela Vanini. BB-SPEEDset: A validated dataset of broadband near-source earthquake ground motions from 3d physics-based numerical simulations. 111(5):2527–2545.

[304] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference.

[305] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. MINE: Mutual information neural estimation.

[306] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5.

[307] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks.

[308] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations.

[309] Bian Li, Hanchen Wang, Shihang Feng, Xiu Yang, and Youzuo Lin. Solving seismic wave equations on variable velocity models with fourier neural operator.

[310] Evan Becker, Parthe Pandit, Sundeep Rangan, and Alyson K. Fletcher. Instability and local minima in GAN training with kernel discriminators.

[311] Lei Huang, Xianglong Liu, Bo Lang, Adams Wei Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks.

[312] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks.

[313] Leslie N. Smith. Cyclical learning rates for training neural networks.

[314] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. Number: arXiv:1711.11586.

[315] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets.

[316] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition.

[317] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

[318] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision.

[319] Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. SoundStorm: Efficient parallel audio generation.

[320] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation.

[321] Alireza Makhzani. Implicit autoencoders.

[322] Fanny Lehmann, Filippo Gatti, Michaël Bertin, and Didier Clouteau. Machine learning opportunities to conduct high-fidelity earthquake simulations in multi-scale heterogeneous geology. 10.

[323] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 347–356. IEEE.

List of Algorithms

List of Figures