

Bachelor Thesis

# Fast Gravitational Waveform Generation for the Einstein Telescope Using GPT-Based Models


*Supervisor:*
Prof. Dr. Alexander Kappes

*Second examiner:*
Prof. Dr. Anton Andronic


*A thesis submitted in fulfilment of the requirements for the degree of*

**Bachelor of Science** *at* Universität Münster

*by*


Hannes Waldeyer

AG Kappes
Institut für Kernphysik


Münster, Dezember 2025

# Contents

**1 Introduction** **1**

**2 Gravitational Wave Astronomy** **3**
2.1 Theoretical Description of Gravitational Waves . . . . . 3
2.2 Gravitaional Wave Detection . . . . . 5
2.2.1 Fundamental Interactions . . . . . 5
2.2.2 Gravitational Wave Interferometers . . . . . 7
2.2.3 Einstein Telescope . . . . . 9
2.2.4 Matched Filtering . . . . . 10

**3 Deep Learning** **13**
3.1 Neural Networks . . . . . 13
3.2 Attention . . . . . 14

**4 Methods** **17**
4.1 Data Generation . . . . . 17
4.2 Architecture . . . . . 18

**5 Results** **21**
5.1 Performance . . . . . 21
5.2 Autoregressive Prediction . . . . . 25

**6 Conclusion** **29**

**A Appendix** **31**
A.1 Convolutional Neural Networks . . . . . 31
A.2 Residual Distributions . . . . . 32

**Bibliography** **35**

**Declaration of Academic Integrity** **39**

# 1 Introduction

Gravity has fascinated people since the existence of mankind. First described as a universal law by Sir Isaac Newton in his book "*Philosophiæ Naturalis Principia Mathematica*", it was 230 years later revolutionized by Einstein in his "theory of general relativity". Being over a century old, the ingenuity of Einstein's theory continues to astonish researchers to this day. The latest contribution to this astonishment was in 2015 with the first detection of a gravitational wave, which once again confirmed Einstein's predictions. Being a direct consequence of general relativity, gravitational waves are emitted by compact heavy objects throughout the universe. Gravitational waves can be understood as a distortion of spacetime propagating at the speed of light, and their effect on particles can only indirectly be seen. This makes the detection of gravitational waves a significant challenge, which can only be overcome by special facilities, built for this purpose. One of those is the Einstein Telescope [1], which is a proposed new generation gravitational wave detector, expected to be operational in the next decade. The detection principle of modern gravitational wave observatories, can for simplicity be described as a kilometer-scaled Michelson interferometer, though in practice, these detectors are by magnitudes more complex and really are engineering masterpieces. In the detectors, the gravitational wave signal is buried in a lot of noise, which makes the use of special algorithms, an inevitable step in the detection process. For signal extraction, algorithms based on "matched filtering" are of key importance. In matched filtering, a known gravitational wave signal is correlated to a number of waveform templates to find suitable parameters, which describe the signal the best. Template banks of current gravitational wave observatories contain millions of different waveforms, which partially need to be generated using numerical relativity [2]. Simulating these waveforms is computationally costly, especially for the merger and ringdown part of the waveform, which can not be approximated using Newtonian methods and therefore the Einstein field equations need to be solved numerically. While generating these templates already is time intensive, the problem is exaggerated for the Einstein Telescope because of its greater sensitivity to different gravitational wave sources, which broadens the parameter space that needs to be covered. A promising approach to tackle this problem is to reduce computation time by generating the merger and ringdown of the waveform with machine learning models, which do not rely on numerical relativity. In this thesis, a generative pre-trained deep learning model will be presented, and its performance on generating gravitational waveforms will be evaluated.

# 2 Gravitational Wave Astronomy

The existence of gravitational waves was first predicted by Albert Einstein in 1916 as a consequence of his work on the general theory of relativity [3].
It took nearly five more decades until 1965 when Joseph Weber built the first experimental setup, the Weber bar, to measure Gravitational radiation [4]. Although these first experiments turned out to not have sufficient precision to detect gravitational waves, they mark the beginning of experimental gravitational wave astronomy. In 1974, R.A. Hulse and J.H. Taylor discovered the binary pulsar PSR 1913+16 [5] which, in subsequent years, provided the first indirect evidence for gravitational radiation through the measurement of its orbital period decay. In the following decades, interferometry emerged as a promising mechanism to directly detect gravitational waves. With time, these detectors got larger and more precise, and in 2015, the Laser Interferometer Gravitational-wave Observatory (LIGO) [6] made the first direct detection of gravitational waves, originating from a binary black hole merger [7].
Following this very short historic introduction, in section 2.1, gravitational waves will be theoretically described, closely following [8, ch. 2.2] and [9, ch. 16] with additional input of [10, ch. 23].

## 2.1 Theoretical Description of Gravitational Waves

In Einstein's general theory of relativity (GR), gravitation is described as the curvature of spacetime, and not as a classical Newtonian force. According to GR, the source of this curvature is the distribution of mass and energy, and their relationship is given by Einstein's field equations also called Einstein Equation. The Einstein equation comprises ten non-linear, partial differential equations and is given by [10, p. 483]

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \frac{8\pi G}{c^4} T_{\mu\nu}, \tag{2.1}$$

where $R_{\mu\nu}$ is the Ricci tensor, $R = g^{\mu\nu} R_{\mu\nu}$ is the scalar curvature, $T_{\mu\nu}$ is the stress-energy tensor and $G$ and $c$ are the gravitational constant and the speed of light, respectively. The left side of eq. (2.1) is also called the Einstein tensor $G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R$ and it only depends on the metric $g_{\mu\nu}$ which represents the general structure of spacetime. To see how gravitational waves are derived from eq. (2.1) the gravitational field is considered to be weak and the observer to be far away from the source $T_{\mu\nu}$ of the gravitational waves. If the gravitational field is weak, then the metric $g_{\mu\nu}$ can be written in the form of

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \, , \tag{2.2}$$

where $\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$ is the Minkowski metric for flat spacetime and $h_{\mu\nu}$ is a small perturbation term ($|h_{\mu\nu}| \ll 1$) whose derivatives are also small. The Einstein tensor $G_{\mu\nu}$ can now be evaluated to linear order in $h_{\mu\nu}$, which together with the

trace-reversed perturbation $\bar{h}_{\mu\nu} = h_{\mu\nu} - \eta_{\mu\nu}h/2$, gives

$$G_{\mu\nu} = \frac{1}{2}\left(\partial_\mu\partial^\alpha\bar{h}_{\alpha\nu} + \partial_\nu\partial^\alpha\bar{h}_{\alpha\mu} - \Box\bar{h}_{\mu\nu} - \eta_{\mu\nu}\partial^\alpha\partial^\beta\bar{h}_{\alpha\beta}\right) . \tag{2.3}$$

Here, $\partial_\mu$ denotes $\partial/\partial x^\mu$, $\partial^\mu = \eta^{\mu\nu}\partial_\nu$ and $\Box = \eta^{\mu\nu}\partial_\mu\partial_\nu$ is the d'Alembertian operator. Equations 2.3 can be further simplified by choosing the Lorenz gauge, which holds the condition $\partial^\nu\bar{h}_{\mu\nu} = 0$. The Einstein tensor then becomes

$$G_{\mu\nu} = -\frac{1}{2}\Box\bar{h}_{\mu\nu} , \tag{2.4}$$

which finally reduces the field Equations eq. (2.1) to

$$\Box\bar{h}_{\mu\nu} = -\frac{16\pi G}{c^4}T_{\mu\nu} . \tag{2.5}$$

This equation is also called the linearized or weak Einstein equation [9, p. 437] and if examined far from the source ($T_{\mu\nu} = 0$), takes the form of the linearized vacuum Einstein equation

$$\Box\bar{h}_{\mu\nu} = \left(-\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \nabla^2\right)\bar{h}_{\mu\nu} = 0. \tag{2.6}$$

The above equation is solved by an Ansatz of the form

$$\bar{h}_{\mu\nu}(t, x, y, z) = A_{\mu\nu}e^{i\left(\vec{k}\cdot\vec{x} - \omega t\right)} , \tag{2.7}$$

which is a plane wave propagating at the speed of light. Additionally, using the transverse-traceless gauge, it can be shown, that for the amplitude tensor $A_{\mu\nu}$ only two of the ten initial degrees of freedom remain. For a gravitational wave propagating in the $z$-direction the amplitude takes the form of

$$A_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & A_+ & A_\times & 0 \\ 0 & A_\times & -A_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = A_+ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + A_\times \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \tag{2.8}$$

A plane gravitational wave therefore consists of a linear combination of two distinct polarizations "plus" (+) and "cross" (×), where $A_+$ and $A_\times$ are the independent amplitudes.

Until now the assumption was that the source of the gravitational waves $T_{\mu\nu}$ is negligible in the far field. Though to understand how gravitational waves are produced, one has to solve eq. (2.3) with the condition $T_{\mu\nu} \neq 0$. This can be done using the general solutions for inhomogeneous wave equations which are known from electrodynamics [9, p. 445]. The solutions can once again be evaluated far from source $T_{\mu\nu}$, where they take the form of

$$\bar{h}_{\mu\nu}(t, \vec{x}) \approx \frac{4G}{rc^2}\int d^3x' T_{\mu\nu}\left(t - \frac{r}{c}, \vec{x}'\right) . \tag{2.9}$$

Here, $r = |\vec{x} - \vec{x}'|$ is the distance between the source $\vec{x}'$ and the observer $\vec{x}$. If the source moves slowly compared to the speed of light (nonrelativistic), the stress-energy tensor $T^{\mu\nu}$ is dominated by the source's rest-mass density $\rho(t, \vec{x})$ and fulfils the conservation of energy and momentum: $\partial_\nu T^{\mu\nu} = 0$ [9, p. 446]. Based on this, one can show the validity of the following relation:

$$\frac{\partial^2}{\partial t^2} Q_{ij} = 2 \int d^3x \, T_{ij} \, , \tag{2.10}$$

where

$$Q_{ij}(t) = \int d^3x \, \rho(t, \vec{x}) x_i x_j \tag{2.11}$$

is the mass quadrupole momentum. The perturbation (eq. (2.9)) can now finally be written as

$$\bar{h}_{ij}(t, \vec{x}) = \frac{2G}{rc^4} \ddot{Q}_{ij}(t - \frac{r}{c}). \tag{2.12}$$

From eq. (2.12) it is obvious that only nonuniform, non-static sources radiate, since $\bar{h}_{\mu\nu}$ depends on the time derivatives of the quadrupole moment and the quadrupole itself is equal to zero for a spherical distribution of mass.
Typical sources of gravitational waves therefore are binary systems of compact heavy objects, e.g. neutron star or black hole binaries, which form a mass quadrupole while rotating around each other. Apart from binary systems, gravitational waves can be emitted by single stars, such as rotating neutron stars that lack perfect spherical symmetry or stars that explode in a supernova.

## 2.2 Gravitaional Wave Detection

In this section, the detection principles of gravitational waves are explained, with a particular focus on laser interferometers, which play a central role in the experiment addressed in this thesis, namely the Einstein Telescope [1].

### 2.2.1 Fundamental Interactions

Understanding gravitational wave detection, first requires analysing how a gravitational wave affects particles. To use the plural here is important because, it is not possible to detect a gravitational wave by the motion of only a single test body. Evaluating the geodesic equation[1] for two particles at rest hit by a gravitational wave, leads to the conclusion that the effect of the gravitational wave leaves the coordinate positions of the particles unchanged but changes the proper distance between them [10, pp. 333-334]. Considering two particles at rest, one at the origin and one at $x = L_*$, $y = z = 0$, if hit by a gravitational wave $h_{xx} = \text{diag}(0, 1, -1, 0) f(t - z)$ propagating in the $z$-direction, the distance between the particles can be written as

[1] The geodesic equation describes the motion of particles on curved spacetime. Further informations can be found here [10, ch. 8].

[10, p. 334]

$$L(t) = \int_0^{L_*} dx \, [1 + h_{xx}(t,0)]^{1/2} \approx L_* \left[1 + \frac{1}{2} h_{xx}(t,0)\right] . \tag{2.13}$$

Following this, the fractional change in distance between the two particles is

$$\frac{\delta L}{L_*} = \frac{1}{2} h_{xx}(t,0). \tag{2.14}$$

This ratio is called the "strain", which is produced by a gravitational wave. For a periodic signal of the form $f(t-z) = A_+ \sin(\omega t + \delta)$ with frequency $\omega$, amplitude $A_+$ and phase $\varphi$, the strain

$$h_{xx} = \frac{1}{2} A_+ \sin(\omega t + \varphi) \tag{2.15}$$

not only changes with time but oscillates with half the gravitational wave amplitude [10, p. 335]. This concept is visualized in fig. 2.1.

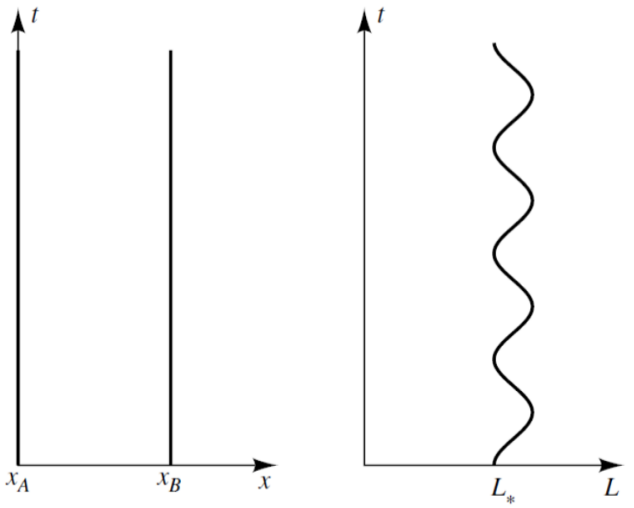



**Figure 2.1:** Motion of two particles located at $x = 0$ and $x = x_B = L_*$, corresponding to the above example. When hit by a wave, the coordinates of the particles do not change (left) but the distance $L$ between them oscillates with the frequency of the wave (right). Taken from [10, p. 335].

Since gravitational waves are transverse waves, the distinct polarization (see eq. (2.8)) can be illustrated with an example similar to the one above. Considering a circular ring of free test particles, a gravitational wave propagating in a perpendicular direction, will deform spacetime in the plane of the ring. This concept can be seen in fig. 2.2. The Gravitational wave first squeezes spacetime along one of the planes axis, while simultaneously stretching it along the other. Half a period later this is repeated but now, spacetime is stretched in the direction in which it was squeezed before. The effect is the same for (+)- and (×)-polarization, though the direction in which spacetime is stretched/squeezed is rotated by 45°.

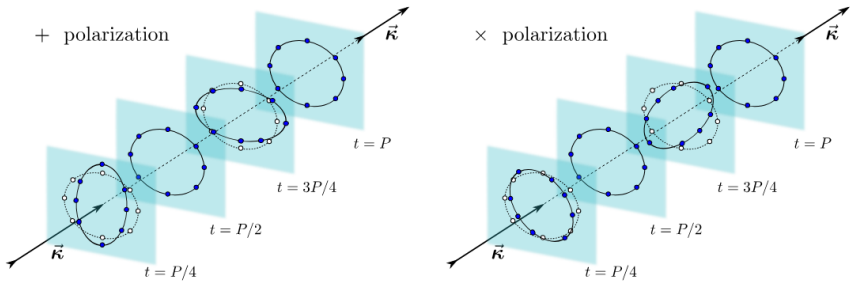



**Figure 2.2:** The two polarizations of a gravitational, illustrated through a ring of particles. Continuous lines and blue dots show the position of the particles at different times, and the dashed lines and white dots show the unperturbed positions. Taken from [11].

The above concepts are the basis for modern gravitational wave detectors, which will be explained in the following section.

### 2.2.2 Gravitational Wave Interferometers

The most basic instrument for understanding the principle of gravitational wave detection is a Michelson interferometer, which is illustrated in fig. 2.3. A Michelson interferometer consists of a monochromatic light source, e.g. a laser, a beam splitter, two mirrors and a detector, for example a photodiode. The outgoing light of the laser, first hits the beam splitter, where part of the light is reflected, and the other is transmitted. The transmitted and the reflected beam then each hit a mirror, which reflects the light back to the beam splitter where, again, a part of each beam is transmitted and reflected. While some light travels back to the laser, the key element is the recombination of the two beams at the photodetector. Since the light of the two beams entered the interferometer at the same time, the interference at the photodetector only depends on the length $\mathrm{L}_X$ and $\mathrm{L}_Y$ of the two arms (see fig. 2.3). For a perfect 50/50 beam splitter and perfectly reflecting mirrors, the power measured by the photodetector can be written as [12, p. 471]

$$|E_{\mathrm{BS}}|^2 = E_0^2 \sin^2(k_{\mathrm{L}}[\Delta L]) \tag{2.16}$$

where $\Delta L = \mathrm{L}_Y - \mathrm{L}_X$ and $k_{\mathrm{L}} = \omega_{\mathrm{L}}/c$ with the frequency $\omega_{\mathrm{L}}$ of the laser. $E_{\mathrm{BS}}$ is the recombined electric field of the beams at the beam splitter. To see how a gravitational wave would interact with the interferometer, the concepts derived in section 2.2.1 are applied. For this, consider the beam splitter as well as the two mirrors to be the test masses affected by a gravitational wave in the transverse traceless (TT) gauge. The interferometers arms are depicted as the $x$- and $y$-axis, with the beam splitter at the origin of the plane. For a "+"-polarized gravitational wave propagating in the $z$-direction, one expects the relative distance between the beam splitter and mirrors to change according to eq. (2.14), while the coordinate positions should stay the same.

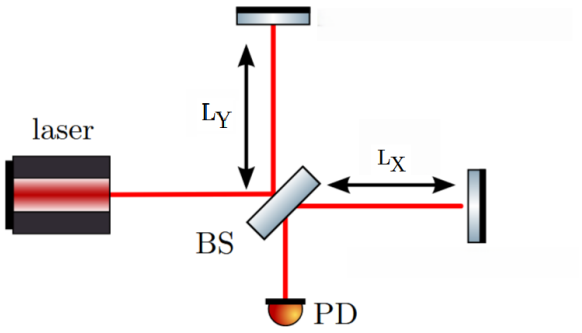



**Figure 2.3:** Schematic representation of a standard Michelson interferometer. The beam splitter is denoted as BS, the photodiode as PD and $L_Y$,$L_X$ are the arm lengths. Taken from [13] (adjusted).

In the TT gauge, the space time interval is given by [12, p. 472]

$$ds^2 = -c^2dt^2 + [1 + h_+(t)]dx^2 + [1 - h_+(t)]dy^2 + dz^2 \,. \tag{2.17}$$

For light travelling along the x-axis from the beam splitter to a mirror at $x = L$, the above reduces to

$$0 = -c^2dt^2 + [1 + h_+(t)]dx^2 \,, \tag{2.18}$$

since for two nearby events on the path of a light ray $ds^2 = 0$ [14]. Equation 2.18 can be used to determine the time $t_1$ it takes for the light to travel from the beam splitter to the mirror $M_1$:

$$t_1 \coloneqq \int_{t_{BS}}^{t_{M1}} dt = \frac{1}{c}\int_0^{L_X} \sqrt{1 + h_+(t)}\, dx \approx \frac{1}{c}\int_0^{L_X} \left[1 + \frac{1}{2}h_+(t)\right] dx \tag{2.19}$$

Similarly, this can be done for the time $t_2$ the light needs for the way back after the reflection at the mirror. Adding $t_1 + t_2 = \tau_x$ results in

$$\tau_x = \frac{2L_X}{c} + \frac{1}{2c}\int_0^{L_X} h_+(t)\, dx - \frac{1}{2c}\int_{L_X}^{0} h_+(t)\, dx \,. \tag{2.20}$$

If the assumption is taken that the wavelength of the gravitational wave is much larger than the interferometers arm length, $h_+(t)$ is nearly constant and eq. (2.20) is simplified to

$$\tau_x = \frac{2L_X}{c}(1 + \frac{h}{2}) \,. \tag{2.21}$$

For light travelling along the $y$-axis, in the same way, one gets [15, p. 44]

$$\tau_y = \frac{2L_Y}{c}(1 - \frac{h}{2}) \,. \tag{2.22}$$

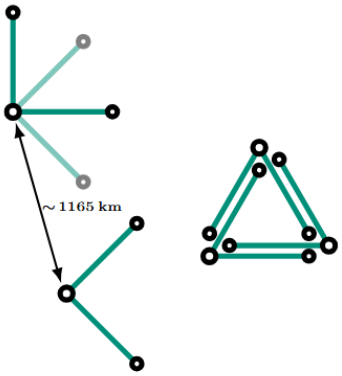
The difference of the sign in front of the $h/2$ in eqs. (2.21) and (2.22) can be understood as the gravitational wave stretching spacetime along the $x$-arm of the interferometer, while simultaneously compressing it along the $y$-arm.
From the above it is obvious, that for $L = L_Y = L_X$, the travel time difference $\Delta\tau = \tau_x - \tau_y$ is not equal to zero. This results in a phase shift between the two beams of

$$\Delta\phi = \frac{4\pi}{\lambda}Lh \tag{2.23}$$

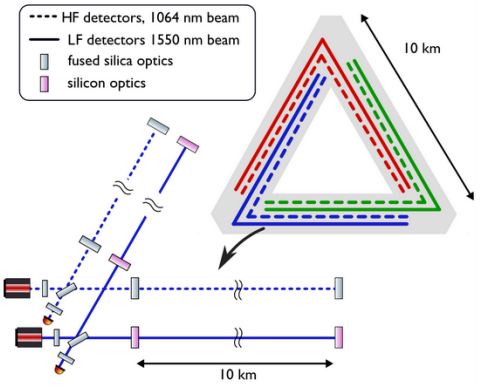
which finally can be measured by the interferometer at the photodetector. Although the above is for a special case, the detection principle is clear: a modulation of power at the photodetector can directly be linked to a modulation of the differential arm length [15, p. 46].
The strain $h$ induced by a gravitational wave is severely small, e.g. of the order $h = 10^{-21}$ for GW150914 [7], the first gravitational wave detected. Consequently, the test masses (beam splitter and mirrors) require sufficient isolation so that the produced strain can be distinguished to phase shifts resulting from noise. Isolation systems can be integrated, e.g. in the mirror suspensions as it is done for aLIGO who features a quadruple-pendulum system [16], passively reducing motion of the mirrors due to noise. Additionally, detectors often have active damping which counteracts seismic noise by a reactive movement of the suspension system [16].

### 2.2.3 Einstein Telescope

The Einstein Telescope (ET) [19] is proposed as a next-generation gravitational wave detector, expected to be operational in the late 2030s. The ET, together with the Cosmic Explorer (CE) observatories [20] and the LIGO Voyager project [21], can be counted as the third generation of terrestrial gravitational wave detectors which are expected to annually detect up to a million gravitational wave events [15, p. 79].
ET will consist of multiple interferometers with an arm length[2] of 10 km to 20 km [17], located at around 250 meters underground.
There are two proposed layouts for the ET. The first layout consists of two L-shaped detectors located several hundred kilometers apart, either arranged in parallel or at angle of 45° to each other (see fig. 2.4a left). The second layout would be comprised of three nested detectors, which are arranged in the shape of a triangle. This layout is also called the "xylophone" layout and can be seen in detail in fig. 2.4b. For the xylophone setup, each detector will consist of two distinct interferometers, where one of the interferometers will be used to detect waves in the low frequency range of 2 to 40 Hz, and the other is devoted




**(a)** Proposed setups for ET. The xylophone setup can be seen at the right and the two variants of the L-shaped setup, to the left. Taken from [17].




**(b)** Detailed overview of the xylophone setup. Each detector will feature two interferometers as can be seen in the lower left. Taken from [18]

**Figure 2.4:** Proposed setups for ET (top) and more detailed triangular layout (bottom).

[2] This depends on the chosen design.

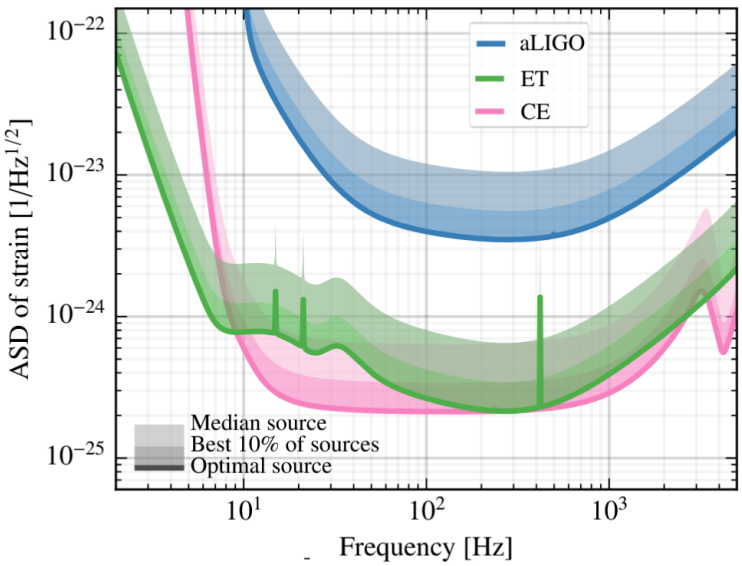



**Figure 2.5:** Amplitude Spectral Densities (ASD) for advanced LIGO (aLIGO), Cosmic Explorer (CE) and the Einstein Telescope. Taken from [19]

to frequencies up to several kHz [15, p. 79]. While each of the proposed layouts performs better or worse for different parameters at different frequencies, it can be expected that ET will feature increased observational capabilities, compared to current detectors like LIGO or VIRGO. The expected sensitivity of the ET can be seen in fig. 2.5 compared to advanced LIGO and the CE observatory, another third generation detector. In the frequency range from 10 Hz to 10 kHz, ET is expected to be sensitive to strain amplitudes roughly an order of magnitude smaller than those accessible to aLIGO, and comparable to those of CE. Moreover, ET will extend the sensitivity to frequencies below 10 Hz, down to about 1 Hz, thereby having better low-frequency coverage compared to CE. Frequencies in the range of 3 Hz to 30 Hz are mainly governed by seismic and thermal noise, which in the xylophone setup will be accounted for by using an additional interferometer with a low power laser and instruments at cryogenic temperatures [19].

### 2.2.4 Matched Filtering

When a gravitational wave is detected, the signal is buried in noise and the waveform is not yet known. To single out the signal from noise, modern observatories use a matched filtering algorithm, which in the following will be briefly explained, based on [22]. Consider a signal $h(t)$ in a data segment of length $[0, T]$ which is buried in noise whose autocorrelation function is $K(t-t')$. A filter $q(t)$ is then defined as the solution of the convolutional integral

$$\int_0^T K(t-t')q(t')dt' = h(t) \tag{2.24}$$

which, expressed in Fourier space, leads to

$$\tilde{q}(f) = \frac{\tilde{h}(f)}{S(f)} \,, \tag{2.25}$$

where $S(f)$ is the power spectral density (PSD). If the filter $\tilde{q}(f)$ is applied to a signal of the form

$$\tilde{x}(f) = \tilde{h}(f) + \tilde{n}(f) \tag{2.26}$$

with additive noise $\tilde{n}(f)$, then the statistic $c$ is obtained

$$c = \int \tilde{q}^*(f)\tilde{x}(f)df \,. \tag{2.27}$$

From this, the signal-to-noise ratio $SNR$ can be defined as

$$SNR = \frac{c}{\sigma_c} = \frac{\int \tilde{q}^*(f)\tilde{x}(f)df}{[\int df \, |\tilde{q}(f)|^2 \, S(f)]^{1/2}} \tag{2.28}$$

where $\sigma_c$ is the standard deviation of c. The matched filter is now defined as the filter, which maximizes the signal-to-noise ratio eq. (2.28). If the SNR reaches a certain threshold, the signal now is considered as a detection of a gravitational wave. Additionally, evaluating the signal in the Fourier space, the parameters of the waveform can be recovered.

# 3 Deep Learning

In classical programming, a machine follows the specific rules of a program. Contrary to that, in machine learning, a machine comes up with its own rules. If one imagines a data processing task, a classical programming solution to the task could be, to look at the given data and come up with an approach that produces the desired output. In machine learning, the goal is to just feed the input data to the system, and let the machine itself find or "learn" appropriate representations of the data, which are meaningful to the task at hand. This ability to derive the rules from the data itself is what makes machine learning powerful and able to learn complex relationships from the training dataset. As "training", the process is described by which the machine learns the specific representations of a dataset. Deep learning is a subfield of machine learning. In deep learning, representations between the input and output are learned in successive layers. The models where these representations are learned in are called "neural networks" and the depth of a model refers to the number of layered representations. A model consisting of two or more successive layers is therefore also called a deep learning model [23].

## 3.1 Neural Networks

The basis of neural networks are simple units called neurons. A Neuron applies different weights $w_i$ to a number of inputs $x_i$ and adds a bias $b$. Additionally, the neuron should not only be able to do an affine transformation of the data, and therefore a non-linear activation function $\sigma$ is applied. For a number of $m$ inputs, the output can be calculated through the following equation:

$$\hat{y} = \sigma\left(b + \sum_{i}^{m} x_i w_i\right) . \tag{3.1}$$

Neural networks consist of an input layer, one or more hidden layers and an output layer. Each layer is made up of several neurons also called nodes. An example of a simple neural network can be seen in fig. 3.1, where each circle represents a node. While the nodes of the input and output layer just forward the given values, the nodes of the hidden layer apply a transformation given by eq. (3.1). The neural network in fig. 3.1 is also called a "deep neural network" since it has more than one hidden layer and all nodes are fully

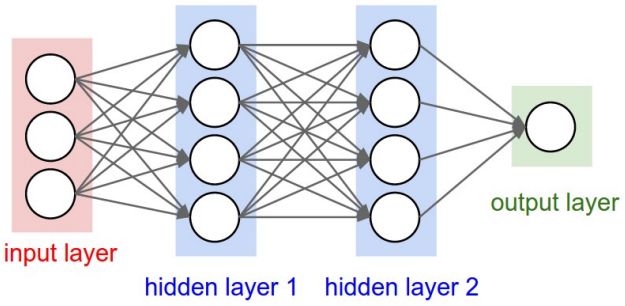



**Figure 3.1:** A simple neural network with one input layer, two hidden layers and an output layer. The flow of computation is denoted by the arrows. Taken from [24].

connected to the nodes of the previous layer [25]. Figure 3.1 is also an example of a "feedforward neural network", where each layer serves as the input to the next layer and there are no loops [25, p. 342]. Using basic linear algebra, eq. (3.1) can be rewritten in a more general form to represent a deep neural network of arbitrary size. If one considers a dataset with $n$ examples, each containing m samples, then the input $\mathbf{X}$ is a matrix of the dimension $n \times m$. Since for each input a weight is applied, the weight matrix $\mathbf{W}$ is of size $m \times d$, where $d$ is the number of neurons in the next hidden layer. Together with the activation function $\sigma$, this leads to the more general equation [25, p. 342]

$$\mathbf{Y} = \sigma(\mathbf{X}\mathbf{W}^{\mathrm{T}} + \mathbf{b}) \,, \tag{3.2}$$

where $\mathbf{b}$ is a vector that applies a bias for each neuron in the hidden layer. In a deep neural network, eq. (3.2) would be applied for each hidden layer.
While the above logic represents the basic structure of a neural network, it still misses a key feature of the "learning" process, which is, the loss function. In "supervised learning" (which is the relevant case for this thesis), the training dataset usually already includes the desired output signals [25]. The loss function, for example, calculates the difference between the output of the network and the "ground truth" values, of the training dataset. This "loss" value can then be used to update the weights and biases of the network, with the goal to minimize the loss of the network. A parameter which impacts how much the weights and biases are updated is the "learning rate" of a network, which the loss is multiplied by. Additionally, different optimization methods can be used, which each implement different strategies on how the loss function is minimized (see [26, ch. 4, ch. 8]).

## 3.2 Attention

This section is based on [25, ch. 16] and [27, ch. 16].

The attention mechanism was first introduced to recurrent neural networks (RNNs). RNNs are designed for processing sequential data in a manner, that parses an entire input sequence, before producing an output. This makes it possible for RNNs to capture the temporal dependencies of a sequence, which can be costly for long sequences. In RNNs, the attention mechanism therefore helps the model to focus on the most relevant input of a sequence. This paradigm was fundamentally shifted in 2017, when a team of Google researchers published a paper called "Attention Is All You Need" [28]. The researchers were able to create a model that was able to capture long-range dependencies among elements in an input sequence, purely based on a self-attention mechanism, not needing RNN layers. The proposed model, was fast to train and appeared to be even more powerful than typical RNNs. The researchers called this new architecture a "Transformer". To understand a Transformer, one needs to understand the attention mechanism which is composed of three stages. To make sense of these three stages, consider an input sequence $x$ and a context vector $z$, both of length $T$, where each element is a vector of dimension $d$. In the first step, weights are calculated which map similarities between the

elements in the input sequence. This can be done using the dot product

$$\omega_{ij} = x_i^{\mathrm{T}} x_j \ , \tag{3.3}$$

which is calculated for each element $i$ and the other elements $j$ within the sequence. In the second step, the calculated weights $w_{ij}$ are normalized using the "softmax" function:

$$\alpha_{ij} = \frac{e^{\omega_{ij}}}{\sum_{j=1}^{T} e^{\omega_{ij}}} = \mathrm{softmax}(\omega_{\mathrm{ij}}) \ . \tag{3.4}$$

The attention weights $\alpha_{ij}$ now sum to one:

$$\sum_{j=1}^{T} \alpha_{ij} = 1 \ . \tag{3.5}$$

Finally, in the last step the attention values in the context vector can be calculated through

$$z_i = \sum_{j=1}^{T} \alpha_{ij} x_j \ . \tag{3.6}$$

While the above calculations are able to map similarities between the input elements, they still lack parameters that can be updated during optimization. To change this, the elements (vectors) of the input sequence $x$ are multiplied by three different weight matrices $W$ to create three different sequences who are called "Query" ($\mathbf{Q} = \mathbf{WX}$),"Key" ($\mathbf{K} = \mathbf{WX}$) and "Value" ($\mathbf{V} = \mathbf{WX}$). The terminology originates from search engines where a typed query is matched to different keys by measuring the relevance of the match with a specific value [23, p. 340]. The three weighted inputs can now be used to calculate the attention values similar to eq. (3.4) but using matrices [27, p. 559]:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{V}, \mathbf{K}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V} \ , \tag{3.7}$$

where $d$ is the dimension of the query and key. The above attention mechanism is called "scaled dot-product attention" and can be expanded to the "multi-head attention" mechanism. Multi-head attention uses multiple query-value-key sets called "heads", instead of just one set. A simple multi-head attention network consisting of three heads $h$ can be seen in fig. 3.2. Before the scaled dot-product attention is applied, the input sequence is mapped to the three different versions $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ using a simple neural network without an activation function (a linear layer) that represents the weight matrix $\mathbf{W}$. The attention is then separately calculated in each attention head, which allows each head, and therefore the model, to attend to different features of the input sequence. After the attention is calculated, outputs of the different heads are concatenated[1] back to a single sequence, which is afterward again linear transformed to match the input dimension.

In fig. 3.2, the attention mechanism consists of a masking option (Mask (opt.))

[1] In computer programming, to concatenate different objects (e.g., tensors) means to stack or join them along an axis, so that a single tensor is created.

which can be applied to the data. Using a mask can limit the input sequence that is passed to the network. This will be explained in more detail in section 4.2.

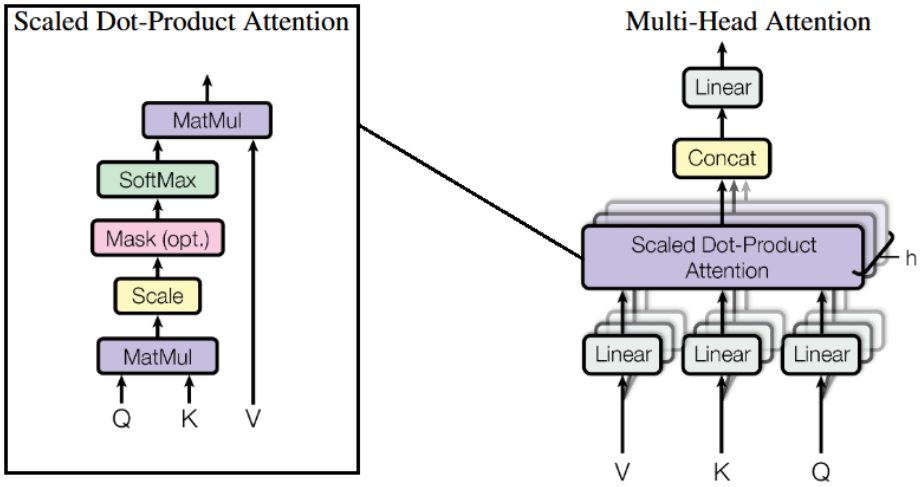



**Figure 3.2:** Right: Multi-head attention mechanism with three heads $h$. Left: Detailed overview of the scaled dot-product attention similar to eq. (3.7) but with a masking option. Adjusted from [28].

# 4 Methods

In this chapter, first the generation of the waveforms in the dataset is explained. Afterward, the used model architecture will be presented.

## 4.1 Data Generation

The dataset used for the training of the model contains around one million (995,328) separate time series, each representing a pure waveform, simulated for ET. The generated waveforms thereby solely represent those of coalescing stellar-mass binary black holes with masses ranging from 10 to 80 solar masses. The code is based on functions from the LIGO Algorithm Library for gravitational-wave analysis (*LALsuite*) [29] and the *PyCBC* project [30]. The data generation process is based on [31] and will be briefly explained in what follows.
First the parameters of the waveform are defined. All waveforms are 16 seconds long and are sampled at a rate of 2048 Hz. Next, the parameters for the specific waveform generation are chosen randomly from given uniform distributions. Besides masses from 10 $M_\odot$ to 80 $M_\odot$, the two black holes can have spins between 0 and 0.998. The distance is fixed at 100 Mpc. Next, the waveform resulting from the black hole binary is simulated using the *SEOBNRv4* model [32]. In general, simulating waveforms with numerical relativity is computationally expensive; for example, generating long waveforms of precessing systems can require up to $\sim 100,000$ CPU-hours [33]. While hybrid models exist which use weak field approximation methods for the inspiral part and only start numerical calculations a few orbits before merger, generating millions of waveforms using these models is beyond the scope of this thesis. This is why the *SEOBNRv4* model is used. The model takes the hybrid approach a step further by also approximating the merger part of the waveforms using the effective-one-body formalism [34], and calibrating the approximated waveforms to simulations based on numerical relativity. Not relying on numerical relativity, these algorithms are significantly faster and match numerical relativity waveforms with accuracies $> 99\,\%$ [32].
Since a detector is only sensitive to a gravitational wave starting at a specific frequency, simulations are started at a cut-off frequency of 4 Hz. This is not physical, and the waveform needs to additionally be tapered at the start. This is done using a one-sided tukey window which the data is multiplied by so it "fades" onto the waveform. In the last step, the waveforms need to be adjusted to how they would be "seen" by a detector. This is done by projecting the waveforms onto the "antenna patterns" of the three ET detectors using "PyCBC" functions. The antenna pattern are functions which take into account that the detector response differs depending on how the detector is oriented to the incident gravitational wave. To calculate the response of the detectors, the antenna patterns rely on the sky location and polarization angle of the source, which require additional parameters that, in the simulation process, are uniformly distributed.

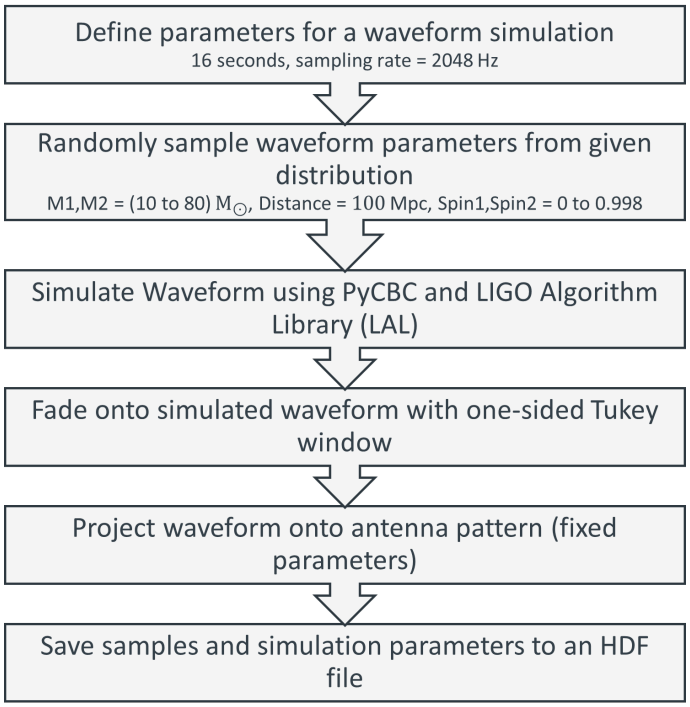



**Figure 4.1:** Flowchart of the waveform generation process. The system is based on [31] and adjusted for ET.

## 4.2 Architecture

The architecture is based on the architecture used in [35] and can be seen in fig. 4.2. The network is a Transformer with causal multi-head attention, featuring three channels corresponding to each detector in the xylophone setup of ET (see section 2.2.3). Training a machine learning model with this architecture on gravitational wave data, the model should eventually be able to generate gravitational waveforms, based partially on its own predictions. This is called an "autoregressive model", where the model generates an output based on some input, extends the input by its output and based on this predicts the next point in time. Autoregressive models are crucial for timeseries forecasting.

The architecture is built up on two layers: The "embedding layer", which prepares the data for the second layer, the "encoder". In advance of the embedding layer, the data is normalized to the maximum strain per channel. Additionally, the input sequences of 32,768 data points is split into multiple non-overlapping patches which are called "tokens". The length of a single token $K$, and consequently the number of tokens per sequence $T$, can vary depending on the chosen model configuration. After the patching, the data thus has the shape of $[T, 3, K]$, being an input sequence of length $T$ where each element is a vector of size $K \times 3$. Lastly, the Fourier transformation of each token is calculated, which makes it possible to simultaneously feed both the time- and frequency-domain information to the model. To increase efficiency, the model accepts a tensor of input sequences rather than just a single waveform, meaning the embedding and encoding layers can process multiple waveforms simultaneously. The number of sequences processed in parallel is defined by the batch size $B$, which is adjustable. Before being passed to the embedding, the

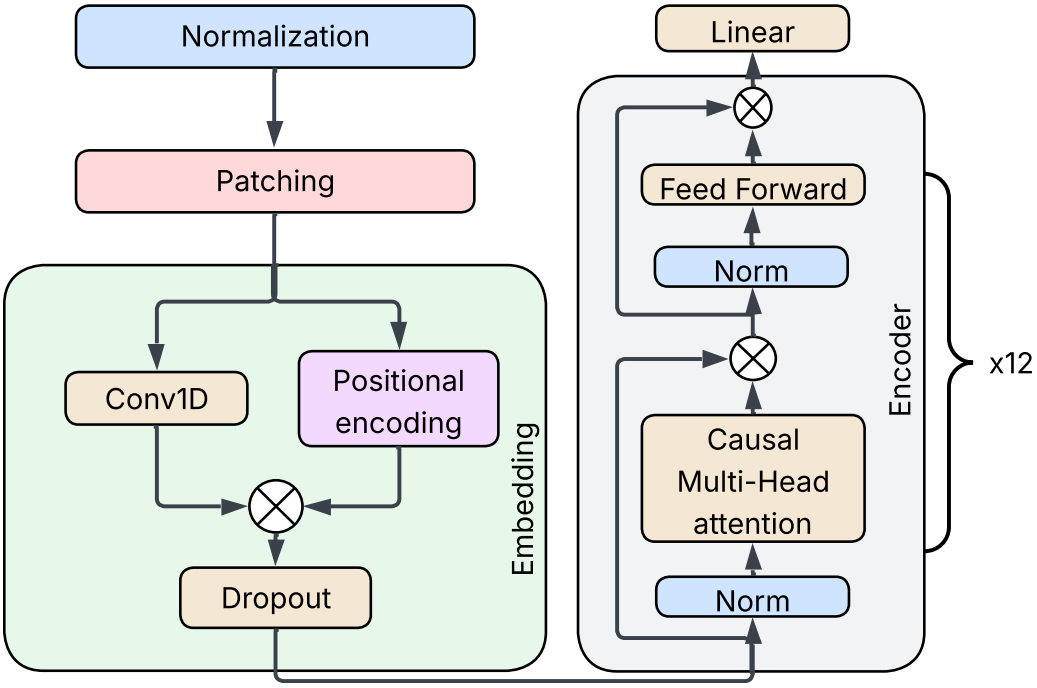



**Figure 4.2:** Fundamental architecture of the used models. Arrows indicate the flow of computation and crosses denote element wise addition.

input dimension needs to be reduced from $[B, T, 3, K]$ to $[B, C \times K, T]$ which is done by flattening[1] the input sequence vectors.

**Embedding:** The key part of the embedding is to prepare the data for the encoder. In the convolutional layers[2] (left), the input data is linearly projected into higher dimensions. This is done in a three layer step, increasing the dimension $d$ of the input vector to 32, 64 and finally 128. As activation function, the "Gaussian Error Linear Unit" (GELU) is used [36]. Projecting the input into a higher dimension enriches the feature representation, which allows the attention mechanism to capture more complex dependencies. While this is great, the attention mechanism does not inherently "memorize" the positional relations of the input data. But, since the positional relation between the input tokens is important when working with timeseries data, positional encoding (PE) needs to be added. In this architecture, sinusoidal PE is applied. The concept of PE encoding is to add a small values to the token of size $d = 128$ at position $pos$, which changes depending on the row $i$ in the input vector according to [25, p. 559]

$$P(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$
$$P(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right) .$$

After the PE is calculated and added to the tokens, an additional layer is applied

[1] Using functions like *torch.flatten*, an input can be reshaped into a one-dimensional tensor.
[2] The convolutional operation is briefly explained in section A.1 of the appendix.

in which some nodes are randomly zeroed (dropped). This is called a dropout layer and stabilizes the models performance.
**Encoder:** After the data is prepared in the embedding module, it is normalized and fed to the attention mechanism of the encoder. In this Transformer architecture, the multi-head attention mechanism is masked. Masking is applied to the $\mathbf{QK}^{\mathrm{T}}$ matrix in eq. (3.7) during the attention mechanism and effectively zeroes the lower triangular part of the $\mathbf{QK}^{\mathrm{T}}$ matrix. This ensures that attention can only be calculated between the current token and past tokens, meaning the model is not able to look ahead. This is called causal attention and important when working on forecasting tasks. The transformer consists of 16 attention heads and a residual connection is applied. After the attention head, the data is again normalized and subsequently fed into a feed forward layer (section 3.1) which uses the GELU activation function. To match the shape of the input data after patching, at the end, the data is linearly projected back.

The specific model parameters for the architecture are listed in table 4.1. Overall, the model comprises 2.5 million trainable parameters, e.g. weights and biases. The number of encoding layers refers to how many identical encoders (right side of fig. 4.2) are stacked. Setting this value to 12 implies that the input embedding is successively transformed through 12 identical encoder layers. As optimization strategy (meaning how the weights are updated during training), a "cosine annealing" learning rate scheduler is used. This scheduler updates the learning rate (LR) of the model based on a cosine function, meaning the LR rises and falls periodically. The minimum LR is usually ten times lower than the initial LR and therefore $5 \cdot 10^{-5}$ for the models used in this thesis. Two other important Hyperparameters are the "stride" and "kernel size" of a model. For additional information on these, the reader is referred to section A.1. For now, it is sufficient to say that the kernel size equals the token length of the input sequence and, since non-overlapping patches are used, also has the same value as the stride. As can be seen in the following section, models with different token length were trained and the kernel size therefore changes depending on the model.

**Table 4.1:** Hyperparameters of the used models. As optimization, a cosine annealing LR-scheduler is used.

| **Parameter** | **Value** |
|---|---|
| Embed. Channels | [32, 64, 128] |
| Encoder Layers | 12 |
| Attention Heads | 16 |
| Batch Size | 128 |
| Initial LR | $5 \cdot 10^{-4}$ |
| Activation $\sigma$ | GELU |

# 5 Results

The model (see section 4.2) was trained for four different token lengths, resulting in a different number of tokens a waveform is split into. These values are listed in table 5.1.

**Table 5.1:** Used token lengths and their corresponding number of tokens per waveform. Each waveform contains 32,768 data points.

| Model | Token length | Number of Tokens |
|---|---|---|
| M32 | 32 | 1,024 |
| M64 | 64 | 512 |
| M128 | 128 | 256 |
| M256 | 256 | 128 |

## 5.1 Performance

During the training, each model's performance is evaluated on an independent dataset called the validation dataset. While the training dataset comprises 90 % of the waveform data and uses this to update the network parameters, the validation dataset makes out the rest of the data and is used to choose the optimal model configuration during training. For this, the mean squared error (MSE) loss between the model's predictions and the ground truth is calculated and the model configuration which minimizes the loss is chosen for the next training cycle. This is done for every training cycle or "epoch", where one epoch is defined as the model's full iteration through the training data. The corresponding loss curves are plotted in fig. 5.1. As can be seen, the validation loss first drops and then converges. This is expected and means that a model has found its best representations of the data and nothing significantly new is learned. Since the training of the model is either stopped after the maximum runtime is reached or the validation loss doesn't improve for ten epochs, this results in a different number of epochs for the four models. To ensure that a model is not solely optimized for the validation and training data, a test dataset is used to evaluate the model's performance after training has finished. The test dataset contains 8,192 separate waveforms, all of which are unseen to the model during the optimization process.
In fig. 5.2 the predictions of each model is plotted together with the corresponding ground truth. Each model tried to predict the same waveform, and the results are shown for the same of the three detectors, denoted as ET2. The amplitude of the waveform (strain) is normalized. Also, the residuals of each sample point are plotted. Since the waveforms are 16 seconds long and generated at 2048 Hz, each plot is about 0.4 seconds long.

In general, it can be seen that the predictions of the models get worse when approaching the merger. While all the models accurately predict the inspiral part

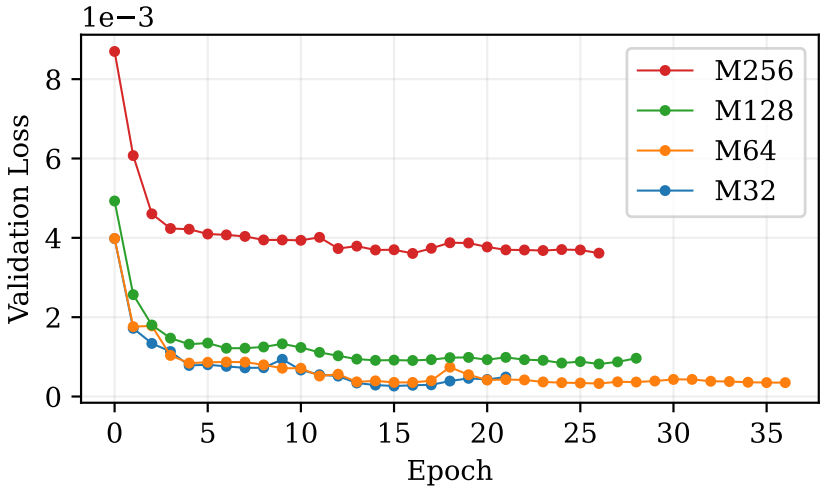
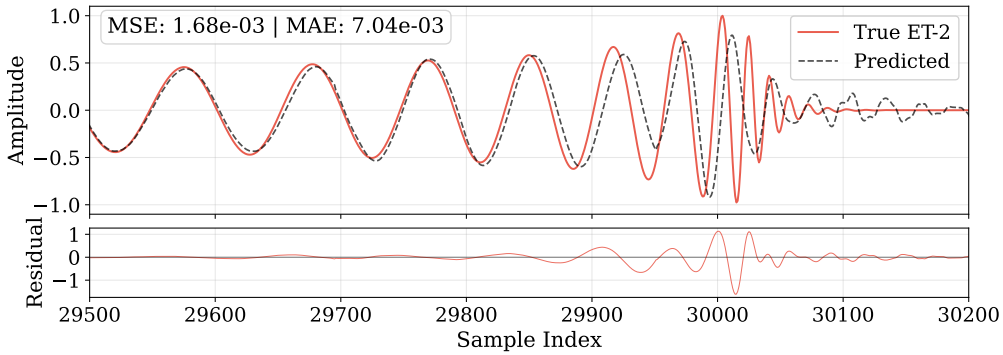



**Figure 5.1:** Learning curve for the four different models.

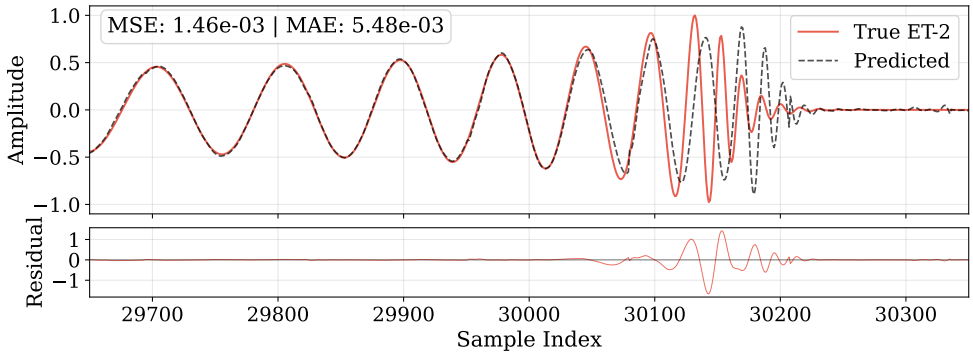
of the waveform, especially the models with a high token length struggle at the late inspiral and merger. As can be seen, the sudden rise in frequency leads to a phase shift, between the predicted and ground truth waveform. While for the models M256 and M128 (see table 5.1), the phase already starts to differ before the merger, it only starts to be apparent for the M64 model at the merger. M32 is the only model, that adapts to the rise in frequency. For the amplitude, the predictions of all models differ from the ground truth, though the predictions of the amplitudes seem to improve with decreasing token lengths. Just as for the frequency, M32 performs the best with a maximum residual of about 0.6.

Looking at the validation loss plots (fig. 5.1) and the predictions of the waveform (fig. 5.2), the assumption is that the prediction error rises with the length of the tokens. This needs to be validated for all waveforms of the test dataset. To do this, the MSE for each waveform in the test dataset is calculated, and the mean is taken. The results for ET2 are shown in fig. 5.3. As can be seen, the error rises with the token length, which agrees with the above observations.

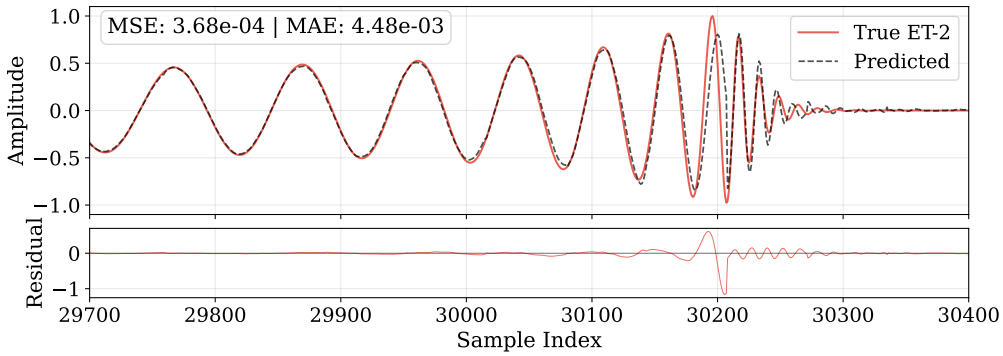
The before plots were only calculated for one detector. To check if there are relevant differences between the predictions of the three ET detectors, the mean Residual of each waveform in the test dataset is calculated, and the distribution is plotted for every detector (fig. 5.4). Figure 5.4 shows the resulting distribution for the M256 model. From fig. 5.4 can be seen, that the distributions of the residuals resemble a Gaussian distribution around $x = 0$. No significant variance between the three detectors can be seen, and mianly, the mean residuals are small. This suggests that the models perform equally well across all three detectors, and therefore, the findings from fig. 5.3 can be generalized.



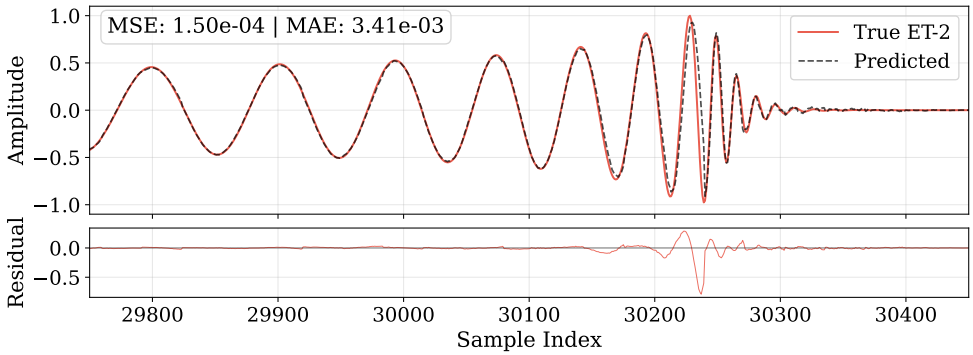

**(a)** M256




**(b)** M128




**(c)** M64




**(d)** M32

**Figure 5.2:** Predictions of the same waveform in ET2, for all the models.

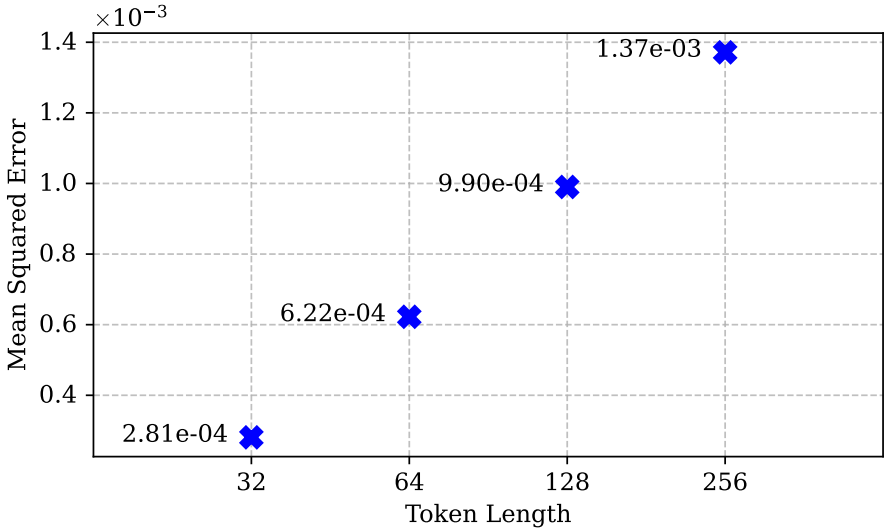



**Figure 5.3:** Mean squared error against token length. The calculated average MSE can be seen alongside the data points. All values were calculated for the detector corresponding to fig. 5.2 that is, ET2.

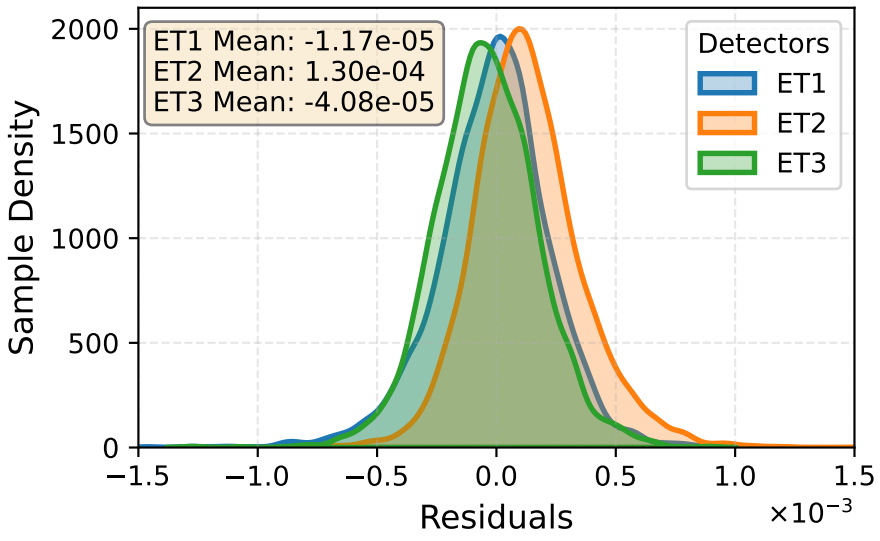
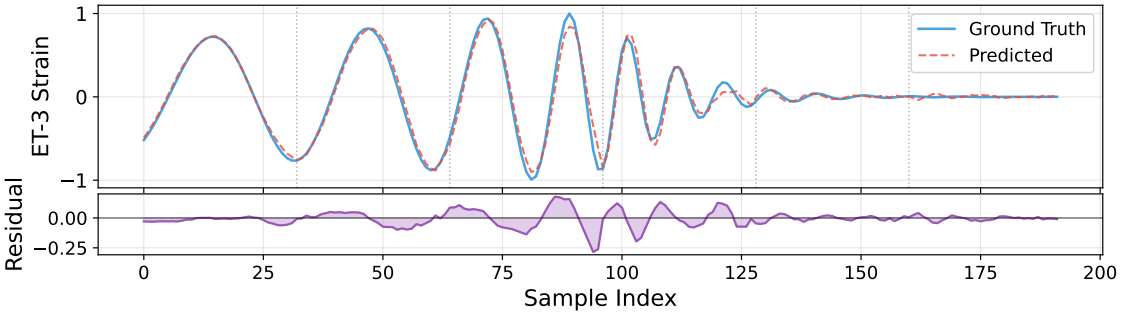



**Figure 5.4:** Distribution of the mean waveform residuals for each detector. The distribution of the other models are shown in the appendix section A.2.

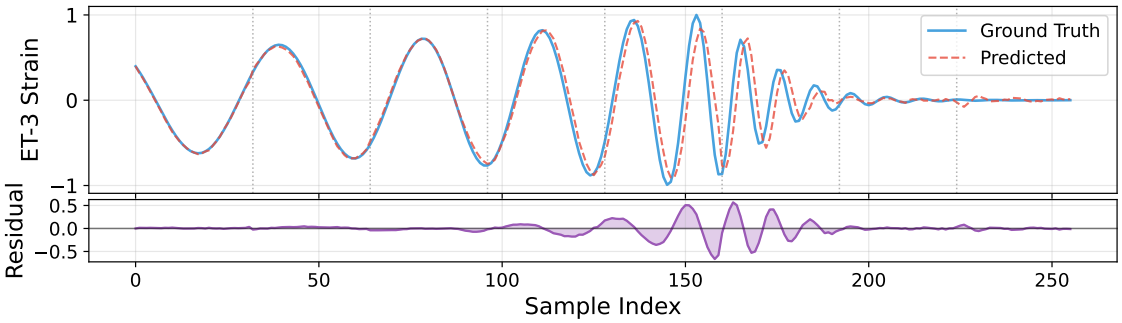
## 5.2 Autoregressive Prediction

For the following section, the masking mechanism introduced in section 4.2 is of key importance. To recap, with a masked attention mechanism, the Transformer calculates attention values for the current token based on all preceding tokens. Therefore, the Transformer is not able to "look" into the future, which is important when dealing with time-series data. Until now, the Transformer solely relied on ground truth data to calculate its predictions. While this is a good way to measure the optimal performance of the Transformer, during waveform generation, there eventually is no "ground truth" preceding the token that the model attempts to predict. In this scenario, the Transformer must use the predictions of prior tokens as its new input context and extend the sequence autoregressively. This strategy, however, introduces a new problem. The prediction for each token inherently contains an error. When the next token is predicted, the error from the prediction of the preceding token is carried along. Consequently, the error accumulates with every generated token. While a small token length previously benefitted the prediction (fig. 5.3), a small token length also implies a larger number of tokens in a sequence (table 5.1), which generally should worsen the autoregressive prediction.
This can be seen in fig. 5.5 for the M32 model. The plots display the merger phase of the waveform, which is autoregressively generated using different sequence lengths. The first token in each sequence is predicted based on the ground truth of all preceding tokens. In fig. 5.5a, six tokens are predicted based on the input of the preceding 860 tokens. As can be seen, the ratio between context token (860) and predicted tokens (6) is rather large, and the model manages to predict the merger with an accuracy similar to the prediction solely based on ground truth data (see fig. 5.2d). In figs. 5.5b to 5.5d, the generated sequence covers more of the inspiral part, and thus the ratio between context tokens and generated tokens is increased. Comparing fig. 5.5a and fig. 5.5b, it can already be seen, that increasing the number of generated tokens prior to the merger by just two, degrades the prediction performance. This is even more evident when increasing the number of generated tokens further, and the error propagation can clearly be seen (fig. 5.5d). Here, the model is not precise enough when adapting to the rising frequency in the inspiral part, and the error made there is carried forward and exaggerated. This results in a phase shift in the prediction of the merger part, while the amplitude still seems to be predicted rather accurately.

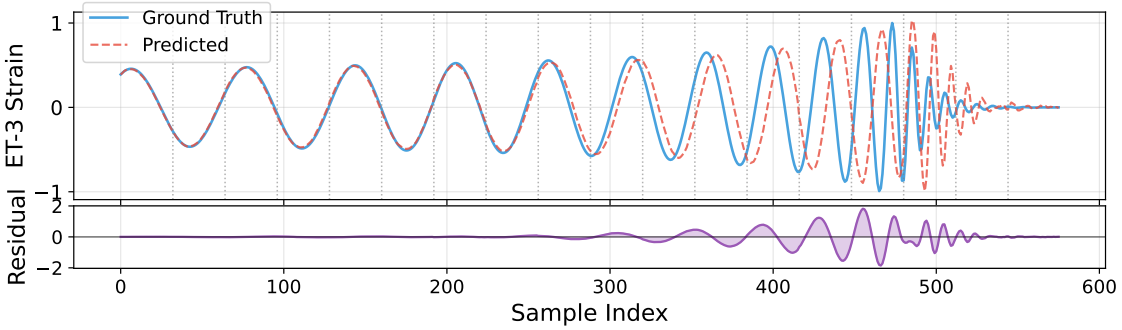
The performance of the M128 model on the long generated sequence can be seen in fig. 5.6. The phase shift prior to the merger is minimal, compared to the phase shift seen in fig. 5.5d. This is the same when comparing the generated sequences of intermediate length fig. 5.5c and fig. 5.7. Although the models with a bigger token length perform better for long generated sequences, none of them are able to, at some point, predict the merger and ringdown as accurately as the M32 model does on short sequences (e.g. fig. 5.5a).



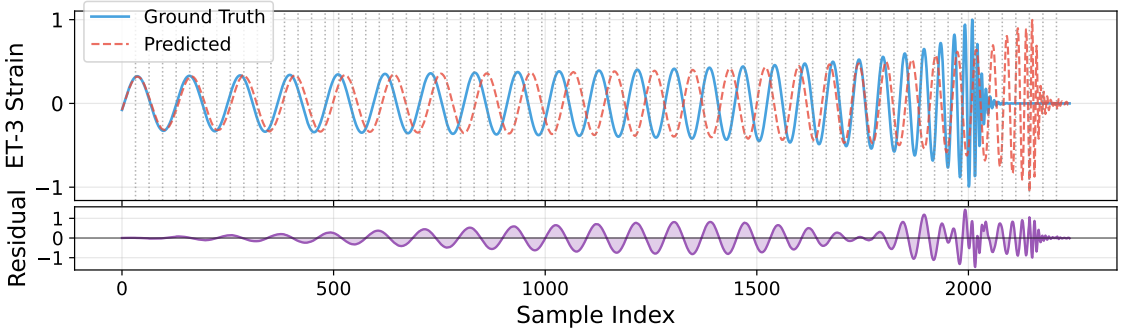

**(a)** Context Tokens: 860, predicted Tokens: 6




**(b)** Context Tokens: 858, predicted Tokens: 8




**(c)** Context Tokens: 848, predicted Tokens: 18




**(d)** Context Tokens:800, predicted Tokens: 70

**Figure 5.5:** Autoregressive predictions of the same waveform in ET3, with the M32 model for different context to prediction ratios.

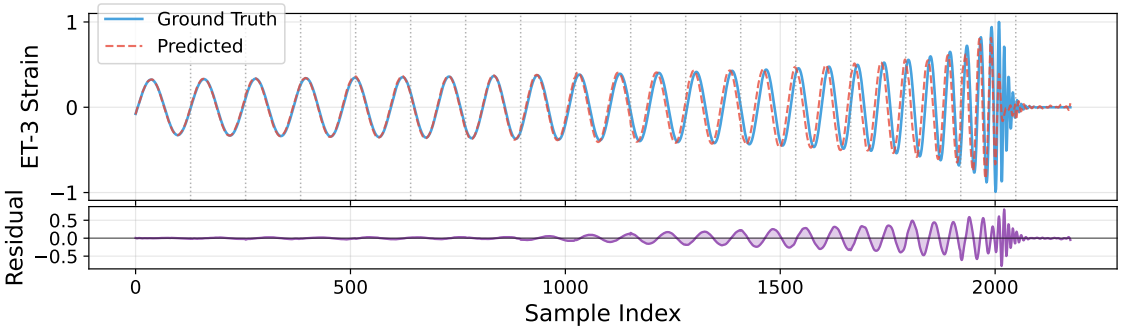



**Figure 5.6:** Generated long waveform for the M128 model. Context Tokens: 200, predicted Tokens:17.

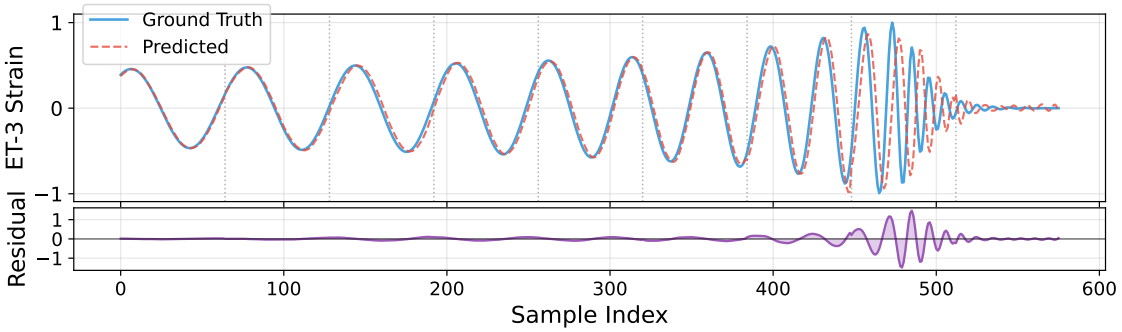



**Figure 5.7:** Waveform generated with the M64 model. Context Tokens: 424, predicted Tokens: 9.

# 6 Conclusion

In this thesis, a deep learning architecture was introduced and optimized regarding the generation of gravitational waveform templates for ET. The architecture is based on a causal multi-head attention mechanism and the models were trained on approximately one million waveforms, which were simulated using *PyCBC* and *LALsuite* functions. Four models were trained, featuring token lengths of 32, 64, 128 and 256 data points.
The validation loss of all models converged during training. The performance of the models was first evaluated on the test dataset using the ground truth data of a single event. Here, the model with the lowest token length (M32) performed the best, with maximum deviations of about 0.6 from the normalized strain at the merger. All other models performed significantly worse. This could also be seen when evaluating the models on the whole test set of 8,192 samples. While the lowest average MSE over all test samples was $1.96e^{-4}$ for M32 model, the MSE is almost ten times this value for the model with the highest token length (M256).
As the primary objective of this architecture is to find faster alternatives to numerical relativity simulations of the merger, the models were tested on generating the merger and ringdown phases based on simulated data of the inspiral part. The testing showed, that the models perform worse the more successive tokens need to be predicted. For the M32 model, starting the generation just two tokens ($\approx 0.044$s) prior to the merger, resulted in maximum residuals of about 0.25. Increasing the number of generated tokens prior to the merger lead to a visible phase shift of the generated merger data, which is caused by erroneous predictions in the generation of the first tokens, which accumulate due to the autoregressive nature of the model. This error lead to merger shifts of up to four tokens (0.063 s) from the ground truth data, when autoregressively generating 64 tokens ($\approx$ 1 s) preceding the merger. While the models with larger token length, expectedly inherit less phase shift when starting token generation way before the merger, none of these model are at some point able to predict the merger and ringdown with accuracies succeeding the M32 when predicting six tokens, starting two tokens before the merger. As suggested by [33] and [37], numerical relativity simulations need to be applied multiple orbits before merger, to produce sufficiently accurate waveforms. The longest generated waveform presented in this thesis is about 13 orbits and the presented models, in this case, either produce a phase shift (M32, M64) or the token length is too high to adjust to the rising amplitude (M128, M256).
To summarize, the M32 model performed best, being able to predict six tokens ($\approx 0.094$ s) of the merger and ringdown with maximum deviation to the normalized strain of 0.25. If the generated waveform is extended, the model's predictions deviate from the ground truth.

# A Appendix

## A.1 Convolutional Neural Networks

The contents in this section are based on [27, ch. 14].
Convolutional neural networks include convolutional layers. Convolutional layers, in difference to fully connected layers, consist of neurons that are not connected to every single input node. These neurons only take the inputs of a number of adjacent neurons in the previous layer, which together are also called the "receptive field". Since the neurons in a convolutional layer have a receptive field of the same size, the overlap of these fields can be determined by a single parameter called "stride". The stride convolutional neural network specifies the shift from one receptive field to the next receptive field. For a 1-Dimensional input and a receptive field size of three, a stride value of three would mean the fields of neighbouring neurons do not overlap. For the same example but a stride value of one, two neurons in the convolutional layer would share one input node.
In convolutional layers, "filters" also called "convolutional kernels" are used to map different features of the input data. The filter represents the weights of a neuron and therefore has to be of the same size as the receptive field. If all neurons of a layer use the same filter, they output a so-called feature map. Using different filters for the same input, will result in multiple feature maps. If the kernel size matches the dimension of the input token, this can also be used to increase the dimensions, in which the data is represented.

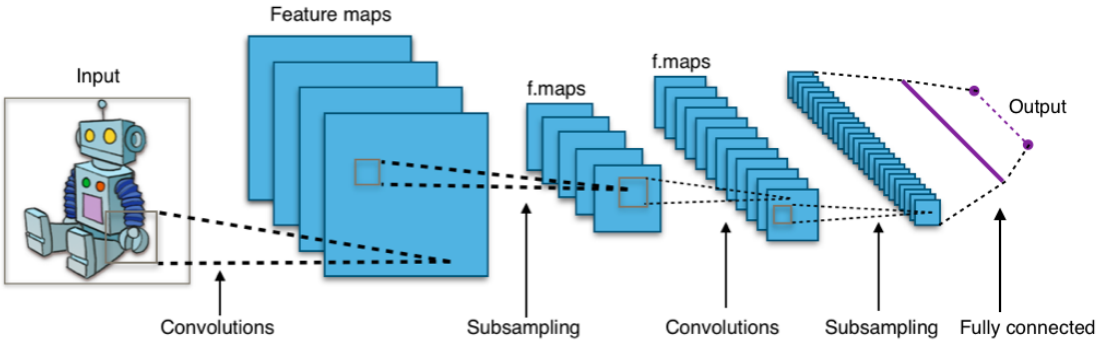



**Figure A.1:** Typical Convolutional Neural Network. Taken from [38].

## A.2 Residual Distributions

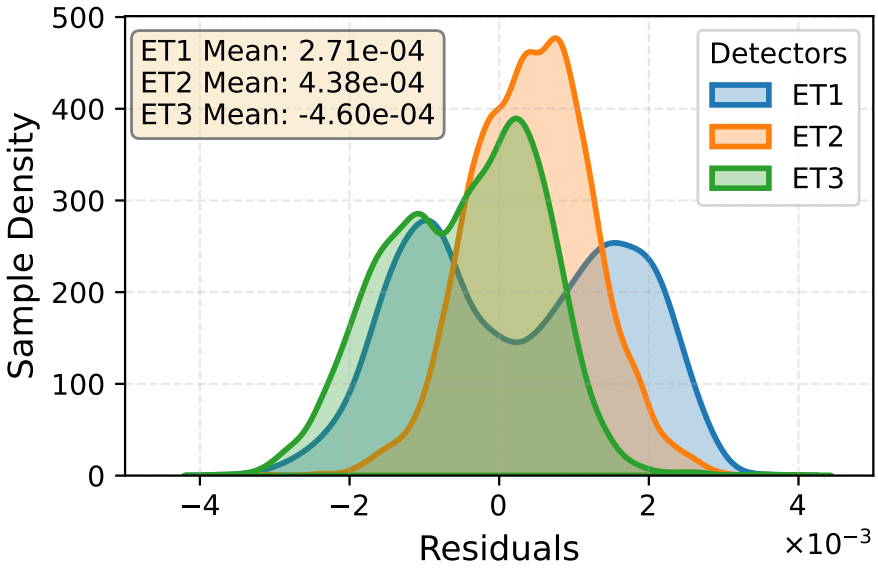



**Figure A.2:** M32

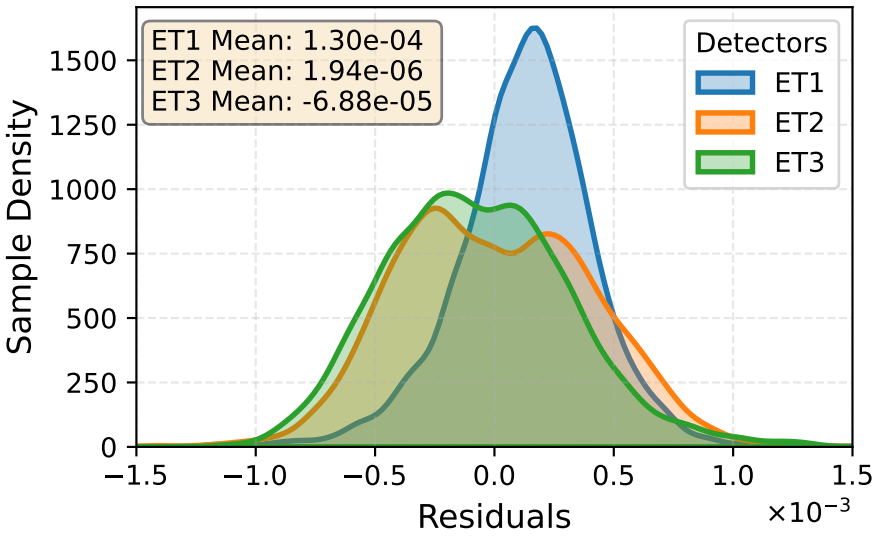



**Figure A.3:** M64

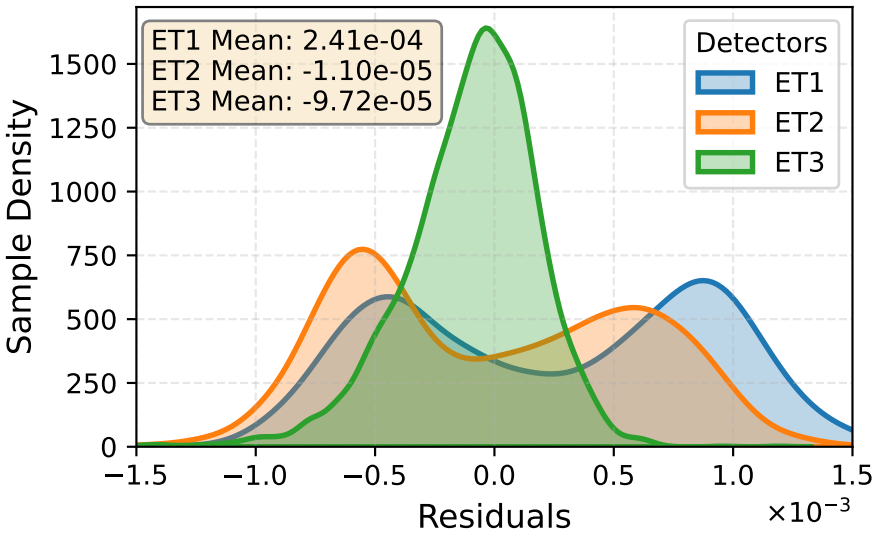



**Figure A.4:** M128

# Bibliography


[1] Adrian Abac et al. *The Science of the Einstein Telescope*. 2025. DOI: 10.48550/ARXIV.2503.12263. URL: https://arxiv.org/abs/2503.12263.

[2] Shio Sakon et al. "Template bank for compact binary mergers in the fourth observing run of Advanced LIGO, Advanced Virgo, and KAGRA" (2022). DOI: 10.48550/ARXIV.2211.16674. URL: https://arxiv.org/abs/2211.16674.

[3] Albert Einstein. *Näherungsweise Integration der Feldgleichungen der Gravitation*. URL: https://adsabs.harvard.edu/pdf/1916SPAW.....688E (visited on 12/23/2025).

[4] J. Weber. "Observation of the Thermal Fluctuations of a Gravitational-Wave Detector". *Physical Review Letters* 17.24 (Dec. 1966), pp. 1228–1230. ISSN: 0031-9007. DOI: 10.1103/physrevlett.17.1228. URL: http://dx.doi.org/10.1103/PhysRevLett.17.1228.

[5] R. A. Hulse and J. H. Taylor. "Discovery of a pulsar in a binary system". *The Astrophysical Journal* 195 (Jan. 1975), p. L51. ISSN: 1538-4357. DOI: 10.1086/181708. URL: http://dx.doi.org/10.1086/181708.

[6] B P Abbott et al. "LIGO: the Laser Interferometer Gravitational-Wave Observatory". *Reports on Progress in Physics* 72.7 (June 2009), p. 076901. ISSN: 1361-6633. DOI: 10.1088/0034-4885/72/7/076901. URL: http://dx.doi.org/10.1088/0034-4885/72/7/076901.

[7] Abbott et al. "Observation of Gravitational Waves from a Binary Black Hole Merger". *Physical Review Letters* 116.6 (Feb. 2016). ISSN: 1079-7114. DOI: 10.1103/physrevlett.116.061102. URL: http://dx.doi.org/10.1103/PhysRevLett.116.061102.

[8] Stephen R. Taylor. *The Nanohertz Gravitational Wave Astronomer*. 2021. DOI: 10.48550/ARXIV.2105.13270. URL: https://arxiv.org/abs/2105.13270.

[9] Michael Ruhrländer. *Allgemeine Relativitätstheorie Schritt für Schritt: Eine Einführung mit Details, Beispielen und Aufgaben*. Springer Berlin Heidelberg, 2021. ISBN: 9783662620830. DOI: 10.1007/978-3-662-62083-0. URL: http://dx.doi.org/10.1007/978-3-662-62083-0.

[10] James B. Hartle. *Gravity: An Introduction to Einstein's General Relativity*. Cambridge University Press, June 2021. ISBN: 9781316517543. DOI: 10.1017/9781009042604. URL: http://dx.doi.org/10.1017/9781009042604.

[11] Nigel T. Bishop and Luciano Rezzolla. "Extraction of Gravitational Waves in Numerical Relativity" (2016). DOI: 10.48550/ARXIV.1606.02532. URL: https://arxiv.org/abs/1606.02532.

[12] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*. Oxford University PressOxford, Oct. 2007. ISBN: 9780191717666. DOI: 10.1093/acprof:oso/9780198570745.001.0001. URL: http://dx.doi.org/10.1093/acprof:oso/9780198570745.001.0001.

[13] Andreas Freise and Kenneth Strain. "Interferometer Techniques for Gravitational-Wave Detection". *Living Reviews in Relativity* 13.1 (Feb. 2010). ISSN: 1433-8351. DOI: 10.12942/lrr-2010-1. URL: http://dx.doi.org/10.12942/lrr-2010-1.

[14] Peter R. Saulson. "If light waves are stretched by gravitational waves, how can we use light as a ruler to detect gravitational waves?" *American Journal of Physics* 65.6 (June 1997), pp. 501–505. ISSN: 1943-2909. DOI: 10.1119/1.18578. URL: http://dx.doi.org/10.1119/1.18578.

[15] *Handbook of Gravitational Wave Astronomy*. 2022. DOI: 10.1007/978-981-16-4306-4. URL: http://dx.doi.org/10.1007/978-981-16-4306-4.

[16] S M Aston et al. "Update on quadruple suspension design for Advanced LIGO". *Classical and Quantum Gravity* 29.23 (Oct. 2012), p. 235004. ISSN: 1361-6382. DOI: 10.1088/0264-9381/29/23/235004. URL: http://dx.doi.org/10.1088/0264-9381/29/23/235004.

[17] "Science with the Einstein Telescope: a comparison of different designs" (2023). DOI: 10.48550/ARXIV.2303.15923. URL: https://arxiv.org/abs/2303.15923.

[18] *Detailed ET xylophone setup*. URL: http://www.gwoptics.org/research/et/layout/ (visited on 12/12/2025).

[19] *ET steering committee, design update 2020*. URL: https://apps.et-gw.eu/tds/?content=3&r=17245 (visited on 12/12/2025).

[20] *Cosmic Explorer project*. URL: https://cosmicexplorer.org/ (visited on 12/17/2025).

[21] Adhikari et al. *LIGO Voyager: White Paper*. LIGO Document T1400226-v12. 2022. URL: https://docs.ligo.org/voyager/voyagerwhitepaper/main.pdf.

[22] Sanjeev Dhurandhar and Sanjit Mitra. *General Relativity and Gravitational Waves: Essentials of Theory and Practice*. Springer International Publishing, 2022. ISBN: 9783030923358. DOI: 10.1007/978-3-030-92335-8. URL: http://dx.doi.org/10.1007/978-3-030-92335-8.

[23] François Chollet. *Deep Learning with Python*. Manning Publications Co., 2021. ISBN: 9781617296864.

[24] Stanford. *CS231n Deep Learning for Computer Vision*. URL: https://cs231n.github.io/convolutional-networks/ (visited on 12/15/2025).

[25] Rashka Sebastian. *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing Birmingham-Mumbai, Feb. 2022. ISBN: 97818081819312.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[27] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Sept. 2019. ISBN: 9781492032649.

[28] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[29] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. *LVK Algorithm Library - LALSuite*. Free software (GPL). 2018. DOI: 10.7935/GT1W-FZ16.

[30] *PyCBC documentation*. URL: https://pycbc.org/pycbc/latest/html/ (visited on 12/12/2025).

[31] Timothy D. Gebhard et al. "Convolutional neural networks: a magic bullet for gravitational-wave detection?" (2019). DOI: 10.48550/ARXIV.1904.08693. URL: https://arxiv.org/abs/1904.08693.

[32] Alejandro Bohé et al. "Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors". *Physical Review D* 95.4 (Feb. 2017). ISSN: 2470-0029. DOI: 10.1103/physrevd.95.044028. URL: http://dx.doi.org/10.1103/PhysRevD.95.044028.

[33] Ilana MacDonald, Samaya Nissanke, and Harald P Pfeiffer. "Suitability of post-Newtonian/numerical-relativity hybrid waveforms for gravitational wave detectors". *Classical and Quantum Gravity* 28.13 (June 2011), p. 134002. ISSN: 1361-6382. DOI: 10.1088/0264-9381/28/13/134002. URL: http://dx.doi.org/10.1088/0264-9381/28/13/134002.

[34] Andrea Taracchini et al. "Effective-one-body model for black-hole binaries with generic mass ratios and spins". *Physical Review D* 89.6 (Mar. 2014). ISSN: 1550-2368. DOI: 10.1103/physrevd.89.061502. URL: http://dx.doi.org/10.1103/PhysRevD.89.061502.

[35] Ruijun Shi et al. *Compact Binary Systems Waveform Generation with Generative Pre-trained Transformer*. 2023. DOI: 10.48550/ARXIV.2310.20172. URL: https://arxiv.org/abs/2310.20172.

[36] *PyTorch Documentation*. URL: https://docs.pytorch.org/docs/stable/index.html (visited on 12/22/2025).

[37] Serguei Ossokine et al. "Multipolar effective-one-body waveforms for precessing binary black holes: Construction and validation". *Physical Review D* 102.4 (Aug. 2020). ISSN: 2470-0029. DOI: 10.1103/physrevd.102.044055. URL: http://dx.doi.org/10.1103/PhysRevD.102.044055.

[38] *Wikipedia: Convolutional Neural Networks*. URL: https://en.wikipedia.org/wiki/Convolutional_neural_network (visited on 12/16/2025).

# Declaration of Academic Integrity

I hereby confirm that this thesis, entitled "Fast Gravitational Waveform Generation for the Einstein Telescope Using GPT-Based Models", is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media and AI Tools, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited. I am aware that plagiarism is considered an act of deception which can result in sanction in accordance with the examination regulations.

Münster, December 25, 2025 _____

I consent to having my thesis cross-checked with other texts to identify possible similarities and to having it stored in a database for this purpose.

I confirm that I have not submitted the following thesis in part or whole as an examination paper before.

Münster, December 25, 2025 _____