GravNet: Detection and Parameter Estimation of Gravitational Waves with Deep Learning

https://github.com/Irwindeep/GravNet

Irwindeep Singh	Ramninder Singh	Sarthak Malviya	Vikrant Singh
B22AI022	B22AI032	B22AI034	B22AI043

1 Introduction

Discovery of the first Gravitational Waves by LIGO/Virgo collaboration opened new paradigms of our understanding of the universe. Contrary to the "Electromagnetic Radiations" based astronomy, gravitational waves are fundamentally unrelated to EM radiation, allowing scientists to observe the universe in a different but complementary perspective. This report presents **GravNet**, a deep learning framework aimed at detecting gravitational wave signals and estimating the associated astrophysical parameters. In this report, we discuss the dataset preparation, the simulation of gravitational waveforms, the design of the different neural network models, and the experimental evaluation of the system.

2 Dataset Preparation and Wave Simulations

The dataset for GravNet was prepared by generating synthetic gravitational waveforms that mimic signals produced by astrophysical events such as binary black hole mergers and neutron star collisions. In the context of gravitational wave detection, data preparation is a critical step. We utilized real LIGO noise data provided by GWOSC and injected simulated waveforms at various SNR (signal to noise ratio) into the noise. Subsequent subsections discuss the steps involved in the same.

2.1 Synthetic Data Generation

So far, only a handful of real gravitational wave events are known which cannot be used for a dataset. To create a realistic dataset, waveform templates were generated using standard models (simulators) in gravitational wave astronomy. We used `SEOBNR_ROM` to simulate Binary Black Hole (BBH) Mergers (which is a reduced order model for `SEOBNR` to save computation) and `TaylorF2` to simulate Binary Neutron Start (BNS) Mergers.

These models utilize parameterized functions where parameters such as mass, spin, and distance can be varied within astrophysically realistic ranges. To reduce computational bottlenecks, we assumed the spins of both masses in the CBC to be zero.

2.2 Noise Injection and Preprocessing

Given that simulated gravitational wave signals are embedded in a real LIGO noise,the synthetic waveforms are scaled to match a selected signal to noise ratio (physically, scaling a wave implies changing the distance of the wave source).

$$\rho_{\text{opt}} = 2\sqrt{\int_0^\infty \frac{|H(f)|^2}{S_n(f)} df}$$

We scale the wave by an amount ρ/ρ_{opt}. Preprocessing steps include normalization of the waveforms and segmentation into fixed-length time-series samples, which are subsequently used

as inputs to the deep learning models. Furthermore, we whitened the injected waveform to have balanced power spectral densities across the signal. Whitening operation is given by:

$$h_{\text{whitened}}(t) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(h(t))}{\sqrt{S_n(f)}}\right)$$

2.3 Wave Simulation Details

Wave simulations were conducted using a combination of numerical relativity and analytical approximations. The simulations account for the dynamical evolution of spacetime curvature in the vicinity of massive bodies. By solving the Einstein field equations under certain approximations, the simulation pipeline produces waveforms that capture the essential physics of gravitational radiation. The resulting dataset not only serves the purpose of training but also provides a testbed for validating the performance of various deep learning architectures. These simulated waveforms form the backbone for training a detection algorithm that must reliably discriminate between noise and actual gravitational wave signals, and for estimating the underlying astrophysical parameters.

2.4 Model Architectures

In the GravNet project, four distinct deep learning models are employed to address both the detection of gravitational waves and the estimation of astrophysical parameters. The design and training of each model are tailored to capture the unique characteristics of the waveform data, as detailed below.

2.4.1 CNN

The Convolutional Neural Network (CNN) architecture serves as the baseline architecture for classification and regression tasks. The CNN is composed of several convolutional layers that learn hierarchical representations of the input waveforms, interleaved with pooling layers to reduce temporal dimensions. Fully connected layers follow the convolutional blocks to produce final predictions. The model is trained using a batch size of 64 over 50 epochs with an Adam optimizer (learning rate set to 1e-3) and cross-entropy loss for classification. This architecture provides a robust starting point for detecting gravitational wave signals and estimating parameters in noisy data.

2.4.2 ResNet

The Residual Network (ResNet) model is detailed in `training_files/train_resnet_cls.py`. The key innovation in ResNet is the introduction of residual (or skip) connections that allow the network to learn residual mappings, facilitating the training of deeper architectures without suffering from vanishing gradients. This model is configured for a single input channel and is adapted to predict multiple parameters (e.g., three astrophysical parameters in this case). With a batch size of 32 and 40 training epochs, the ResNet is trained with cross-entropy loss for classification. For the regression task of parameter estimation, the network's performance is evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). The residual blocks ensure robust feature extraction and improved convergence, particularly when dealing with complex, noisy gravitational wave signals.

2.4.3 DenseNet

The DenseNet model, as implemented in `training_files/train_densenet_cls.py`, leverages dense connectivity principles. In DenseNet, each layer receives input from all preceding layers, which encourages feature reuse and mitigates the vanishing gradient problem even in very deep networks. The architecture, defined in `gravnet/densenet.py` via the `DenseNet` module, is designed to capture fine-grained features from the waveform data. Similar to the CNN and ResNet architectures, DenseNet is trained for both classification (using cross-entropy loss) and regression

(using MSE and MAE metrics) tasks. The dense inter-layer connectivity improves learning efficiency and helps the model to extract subtle patterns embedded in the noisy gravitational wave signals.

2.4.4 UNet

The UNet architecture is adapted from its original application in biomedical image segmentation to serve a dual purpose in GravNet—first for noise extraction and then for wave detection and parameter estimation. The training of the UNet model is performed in two stages:

1. **Pretraining on Noise Extraction:** In `training_files/train_unet.py`, the UNet is first pretrained using a dedicated noise extraction dataset (`NoiseExtData`). This dataset simulates noise profiles encountered in gravitational wave observations, and the pretraining phase helps the model learn to effectively separate signal from background noise. Pretraining MSE Loss is 62.9203 and MAE Loss is 2.98.

2. **Adaptation for Wave Detection and Parameter Estimation:** After pretraining, the UNet is fine-tuned to detect gravitational waves and estimate key parameters. The network architecture is defined with encoder channels set as `[1, 32, 64, 128, 256, 512]` and a kernel size of 3. The symmetric encoder-decoder structure, along with skip connections, enables the model to maintain high-resolution features critical for localizing transient signal components. The fine-tuning stage employs Mean Squared Error (MSE) as the loss function and uses an Adam optimizer with a learning rate of 1e-3, training over 40 epochs with a batch size of 64.

Each of these architectures is designed to address specific challenges associated with gravitational wave data—namely, the detection of faint signals buried in noise and the accurate estimation of astrophysical parameters. Their implementations, as provided in the corresponding training scripts, highlight the experimental rigor and adaptability of the GravNet framework in tackling real-world problems in gravitational wave astronomy.

3 Experimental Setup and Results

The experimental evaluation of the GravNet framework involves assessing both the detection accuracy of gravitational waves and the precision of parameter estimation. The models were trained on the simulated dataset described earlier, and the training process utilized common optimization algorithms such as Adam with learning rate decay schedules. The training data was split into training, validation, and test sets to ensure unbiased evaluation.

For the detection task, the models output binary classifications where the presence of a gravitational wave is marked as a positive detection. For the parameter estimation task, regression outputs corresponding to key astrophysical parameters (e.g., masses, spins, distance) are produced. The evaluation metrics for detection include accuracy, cross entropy loss, while the regression task is assessed using Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Table 1: Performance Metrics

Model	Wave Detection Accuracy (%)	CE Loss	MSE	MAE
CNN	98.45	0.0606	113.44	8.62
ResNet	95.87	0.3179	121.29	5.97
DenseNet	98.07	0.0751	190.36	7.81
UNet	**98.83**	**0.0517**	**29.73**	**2.98**

4 Conclusion

In this report, we have presented GravNet, a deep learning framework for the detection and parameter estimation of gravitational waves. Starting from the generation of a realistic synthetic dataset, we described the methods used to simulate gravitational waveforms and inject realistic noise. The core of the project lies in the application of various deep learning architectures including CNN, ResNet, DenseNet, and UNet, each contributing unique strengths to the task of signal detection and feature extraction.

Table 2: Contributions

Irwindeep Singh	Dataset Handling, UNet implementation and adaptation
Ramninder Singh	CNN Implementation
Sarthak Malviya	ResNet Implementation
Vikrant Singh	DenseNet Implementation