



Data efficient machine learning for gravitational wave localization

Yuya Hashimoto, Kazuharu Bamba
Fukushima University

- In recent years, analyzing gravitational waves using machine learning has received much attention. However, machine learning typically requires large datasets, which is computationally expensive.
- In this study, we show that a four-detectors including KAGRA achieves higher accuracy than the three-detectors (LIGO H1, LIGO L1, and Virgo), even with a small dataset.
- Furthermore, we demonstrate that we can maintain over 80% of the performance of the previous study [S. Sasaoka et al., Phys. Rev. D **105**, 103030 (2022)] using only one-tenth of the data, achieved through data augmentation and TCN optimization.

Introduction

- Gravitational waves from compact binary mergers were first detected in 2015 [B. P. Abbott et al., Phys. Rev. Lett. **116**, 061102 (2016)], and can be a new method of observation. Analyzing data containing various types of noise is required to detect the waves and quickly determine gravitational wave source.
- A previous study [S. Sasaoka et al., Phys. Rev. D **105**, 103030 (2022)] employed machine learning for localization and demonstrated the improvement in accuracy by adding KAGRA to the LIGO H1, LIGO L1, and Virgo. However, deep learning typically requires massive datasets.
- In this work, we show that a data-efficient TCN model can use only one-tenth of the training data and still maintain over 80% performance.

Methodology

I. Data generation process

- We generate data using the open-source tool ggwd [T. Gebhard et al., Phys. Rev. D **100**, 063015 (2019)]. Since the tool only generates GW strain data from LIGO H1 and LIGO L1, we edit to generated data from Virgo and KAGRA.
- The waveform parameters used for data generation are shown in the table I. This followed previous study [C. Chatterjee et al., Phys. Rev. D **100**, 103025 (2019)].
- We generate 20,000 samples for training, 4,000 for validation, and 4,000 for testing. This is one-tenth the size of the previous study [S. Sasaoka et al., Phys. Rev. D **105**, 103030 (2022)].

Table I. Parameters used to simulate

Parameter	Prior Range
Mass 1, Mass 2	$[30M_{\odot}, 80M_{\odot}]$
Spin 1z, Spin 2z	$[0, 0.998]$
Right ascension	$[0, 2\pi]$
Declination	$[-\pi/2, \pi/2]$
Coalescence phase	$[0, 2\pi]$
Inclination	$[0, \pi]$
Polarization	$[0, 2\pi]$
SNR	$[10, 50]$

II. Data augmentation

- We apply random time shifts and noise injection to the generated training samples. This data augmentation technique is inspired by previous study [Iwana and Uchida, arXiv:2007.15951v4].
- **Random time shifts** : We train the model to be invariant to temporal shifts by randomly altering the peak positions of signals.
- **Noise injection** : We inject Gaussian noise with a dynamic SNR. This suppresses overfitting and improves generalization performance.

III. Temporal Convolutional Networks

- TCN [S. Bai et al., arXiv:1803.01271v2] is a model used for sequential data modeling. It consists of several temporal blocks with extended causal convolutions and residual connections. The network structure is shown in the Fig. 1.
- We stack seven temporal blocks with a kernel size of 7.

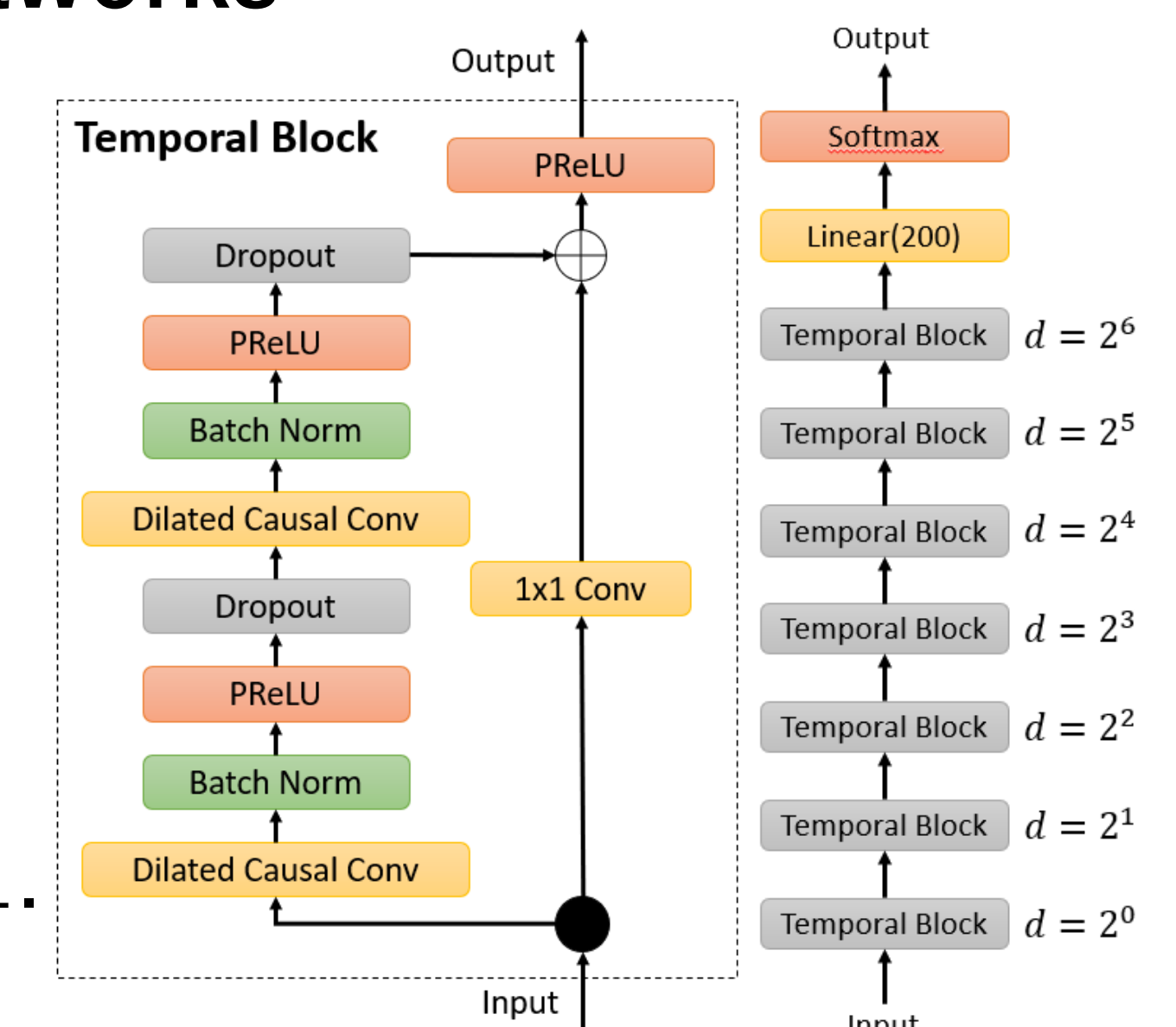


Fig. 1. TCN architecture

Results

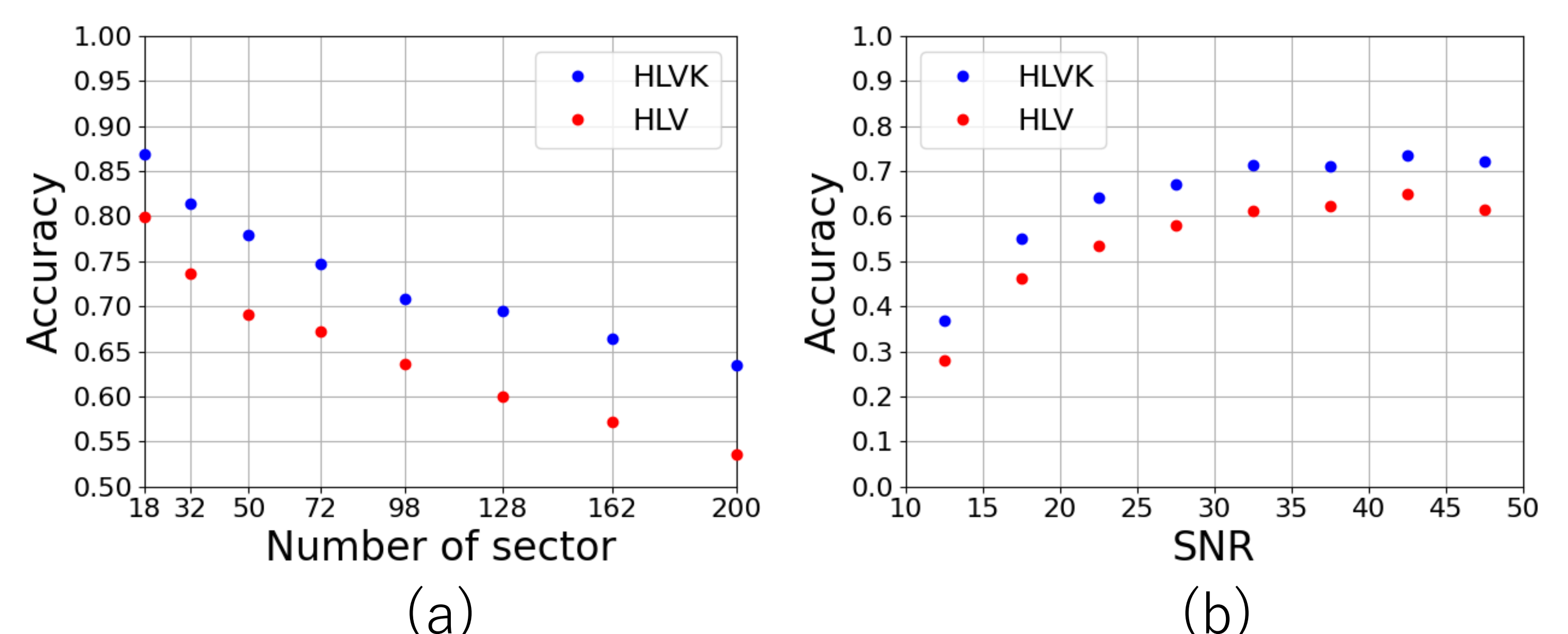


Fig. 2. (a) The localization accuracy using three (LIGO H1, L1 and Virgo) and four (LIGO H1, L1, Virgo, and KAGRA) detectors. (b) The localization accuracy for 200 sectors vs SNR.

- As shown in Fig. 2(a), accuracy improves as the number of detectors increases. These results are similar to a previous studies [S. Sasaoka et al., Phys. Rev. D **105**, 103030 (2022)] and show that fourth detector is highly effective when the number of sectors is large.
- As shown in Fig. 2(b), this model is unable to maintain accuracy with low SNR data. However, using a fourth detector improved the accuracy.

Conclusions

- Even if only one-tenth of the training data compared to previous study [S. Sasaoka et al., Phys. Rev. D **105**, 103030 (2022)], we have demonstrated that our TCN model with data augmentation maintained over 80% classification accuracy.
- In the future, we plan to continue optimizing the TCN to improve robustness and generalization performance.
- One future task is to compare the results with those obtained from ensemble learning approaches using multiple TCNs.