Deepclean: Denosing of Gravitational Wave using Machine Learning

Chia-Jui Chou
National Yang Ming Chiao Tung University, Taiwan

2022/01/25
GW Denoising

- GW signals are small compared to the background noise.
- Proper denosing will increase the Signal-to-Noise Ratio (SNR) and the sensitivity of the detector which gives better ability to detect events and improve the accuracy of parameter estimation.
- Independent Component Analysis (ICA) is designed for the denoising in data analysis.
- Deepclean is designed to do the denoising for the low-latency online search.
Deepclean and Low-latency GW Search

- The detection of the GW of BNS or CCSN provides the early warning for the follow-up detection of X-ray and neutrino.
- Fast denoising using Deepclean will increase the sensitivity in the low-latency GW search.

![Graph showing Inspiral, Merger, and Ringdown](image)
Calibration and Strain

- Variation of $h(t)$:
  - C00 (Front-End Calibration): generated in the digital control,
  - C10 (C00 in LIGO, Virgo): generated by the Low-Latency pipeline of calibration (Finite Impulse Response (FIR) filters based on gstlal library),
  - C20 (C10 in LIGO, Virgo): generated by offline calibration several months after the data acquisition (gstlal-based).
Deepclean and ICA

- ICA is used to do the denoising of C20 strain.
Deepclean and ICA

- ICA is used to do the denoising of C20 strain.
- Deepclean is designed to do the fast denoising of low-latency C10 strain.
Deepclean: Denoising using PEM and Auxiliary Channels

- \( w(t) \): witness channels that record the data from the auxiliary sensors and PEM.
- \( p(t) \): predicted noise in strain.
- Both \( w(t) \) and \( p(t) \) are 8-second long timeseries with the same sampling rate.
- Ref: 2005.06534, 2108.12430

Encoder (down-sampler)  Decoder (up-sampler)
Deepclean: Training and Inferencing

- $h(t)$: raw strain data.
- $r(t) = h(t) - p(t)$: residual.
- Loss function: $J_{asd} = \frac{1}{M} \sum_{i=0}^{M-1} \sqrt{\frac{S[r_i, r_i]}{S[h_i, h_i]}}$. 

![Diagram of Deepclean model showing the flow of data from $w(t)$ through a series of Conv1d layers, a latent vector, and back through Transpose Conv1d layers to produce $p(t)$, with $h(t)$ and $r(t)$ as inputs and outputs.](image)
Deepclean using 2 PEM Channels

- **O3GK run:**
  - 2020/04/07-08:00:00 UTC (1270281618) ~ 2020/04/21-00:00:00 UTC (1271462418)
  - Calibration parameters was switched on: 2020/04/15 23:00:30 UTC (1271026848)

- **Strain data used:**
  - K1:DCS-CALIB_STRAIN_C10

- **PEM channel used:**
  - K1:PEM-MIC_PSL_TABLE_PSL4_Z_OUT_DQ
  - K1:PEM-SENSOR_RACK_OMC1_DSUB2_OUT_DQ

- **Frequency Filter:** 55Hz ~ 65 Hz
- **Training Time:** 1270299936 ~ 1270300064 (128 second)
- **Cleaning Time:** 1270300064 ~ 1270300192 (128 second)
Deepclean using 2 PEM Channels: Raw Strain

Raw Strain: C10

Raw Strain: C20
Deepclean using 2 PEM Channels: C20 State Vector

<table>
<thead>
<tr>
<th>bit</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HOFT_OK</td>
<td>h(t) was successfully computed</td>
</tr>
<tr>
<td>1</td>
<td>OBS_INTENT</td>
<td>The “observation intent” is set by the operator</td>
</tr>
<tr>
<td>2</td>
<td>OBS_READY</td>
<td>Interferometer is in “observation ready” mode</td>
</tr>
<tr>
<td>3</td>
<td>(not in use)</td>
<td>(not in use)</td>
</tr>
<tr>
<td>4</td>
<td>NO_GAP</td>
<td>The input data was present (not a gap)</td>
</tr>
<tr>
<td>5</td>
<td>NO_STOCH_HW_INJ</td>
<td>No stochastic hardware injections</td>
</tr>
<tr>
<td>6</td>
<td>NO_CBC_HW_INJ</td>
<td>No compact binary coalescence hardware injections</td>
</tr>
<tr>
<td>7</td>
<td>NO_BURST_HW_INJ</td>
<td>No burst hardware injections</td>
</tr>
<tr>
<td>8</td>
<td>NO_DETCHAR_HW_INJ</td>
<td>No detector characterization hardware injections</td>
</tr>
</tbody>
</table>
Deepclean using 2 PEM Channels: C20 Data Quality

Data Quality Flags: 1270299904 – 1270300224
Deepclean using 2 PEM Channels: C20 Data Quality
Deepclean using 2 PEM Channels: PEM Channels

\[ w(t) \]
Deepclean using 2 PEM Channels: Predicted Noise

\[ p(t) \]
Deepclean using 2 PEM Channels: Predicted Noise

\[ p(t) \]
Deepclean using 2 PEM Channels: ASD of PEM Channels
Deepclean using 2 PEM Channels: ASD of the Strain
Deepclean using 2 PEM Channels: ASD of the Strain

![Graph showing ASD-C20 comparison]

- **Y-axis**: $\text{ASD [Hz}^{-1}\text{]}$,
- **X-axis**: Frequency (Hz),
- **Legend**:
  - C20-DC
  - C20-RAW
Comparison to Independent Component Analysis (ICA)

From Jun'ya Kume's talk in 7th KAGRA International Workshop, 2020/12/19
Comparison to Independent Component Analysis (ICA)

- We can see the 60Hz peak and the 58.68Hz peak in the ASD of C10 strain are decreased by the Deepclean.
- Further optimization of Deepclean needs to be done.
- ICA has better subtraction on the 58.68Hz peak.
Summary

- Deepclean uses the data from the auxiliary sensors and PEM to predict the noise $p(t)$ in the strain $h(t)$. The cleaned strain: $h(t) - p(t)$.
- Deepclean aims to do fast online denoising to increase the sensitivity of the detector in the low-latency search of GW.
- We have shown that Deepclean has the ability to subtract the 60Hz and 58.68Hz noise captured by the PEM channels.
- Low-frequency (8Hz ~ 22Hz) denoising is essential in early detection, which noise sources and related PEM channels we need to look at?
Thank you for the attention
Appendix
Amplitude Spectral Density

From Nami Uchikata (ICRR), Kideyuki Tagoshi (ICRR), Re-computing of O3GK BNS range, Detchar meeting 2020/01/28
Calibration and Strain

- The strain:
  \[ h(t) = \frac{L_x(t) - L_y(t)}{L_0}, \quad L_0 = 3000\text{m in KAGRA}. \]

- Photon calibrator (PCAL) and gravity field calibrator (GCAL) are used for precise calibration.

- Variation of \( h(t) \):
  - C00: generated in the digital control,
  - C10: generated by real time pipeline of calibration (Finite Impulse Response (FIR) filters using goslal),
  - C20: generated by offline calibration several months after the data acquisition (goslal).
PEM in KAGRA

KAGRA PEM SENSOR LOCATIONS
Ver. 2020/4/7
O3GK
Amplitude Spectral Density

1. Pick a Fourier Transform Length T (usually 8 second).
2. Multiply h(t) with a window function (hann) to avoid spectral leakage.
3. Apply Fourier transform to h(t): \( \tilde{h}(f) = \int_{-T/2}^{T/2} h(t) e^{-2\pi i ft} dt. \)
4. Estimate the power of h in frequency domain: \( \tilde{h}^*(f) \cdot \tilde{h}(f). \)
5. Do the estimation above in many different time segments and get the average of all segments to get the power spectral density (PSD) of h(t):
   \[
   \langle \tilde{h}^*(f) \cdot \tilde{h}(f) \rangle = \frac{T}{2} S[h, h](f) \rightarrow S[h, h](f) = \frac{2}{T} \langle \tilde{h}^*(f) \cdot \tilde{h}(f) \rangle.
   \]
6. Take the square root of the PSD to get the amplitude spectral density (ASD):
   \[
   \sqrt{S[h, h](f)}.
   \]
Denoising of 60Hz Powermain Noise

- **O3GK run:**
  - 2020/04/07-08:00:00 UTC (1270281618) ~ 2020/04/21-00:00:00 UTC (1271462418)
  - Calibration parameters was switched on: 2020/04/15 23:00:30 UTC (1271026848)

- **Strain data used:**
  - K1:DCS-CALIB_STRAIN_C10
  - K1:DAC-STRAIN_C20

- **PEM channel used:**
  - K1:PEM-SENSOR_RACK_OMC1_DSUB2_OUT_DQ

- **Frequency Filter:** 55Hz ~ 65 Hz

- **Training Time:** 1270299936 ~ 1270300064 (128 second)

- **Cleaning Time:** 1270300064 ~ 1270300192 (128 second)
Denoising of 60Hz Powermain Noise: C10 (Low-latency)
Denoising of 60Hz Powermain Noise: C10 (Low-latency)
Denoising of 60Hz Powermain Noise: C10 (Low-latency)
Denoising of 60Hz Powermain Noise: C10 (Low-latency)
Denoising of 60Hz Powermain Noise: C20

\[ p(t) \]
Denoising of 60Hz Powermain Noise: C20
Denoising of 58.68Hz using Microphone

- **O3GK run:**
  - 2020/04/07-08:00:00 UTC (1270281618) ~ 2020/04/21-00:00:00 UTC (1271462418)
  - Calibration parameters was switched on: 2020/04/15 23:00:30 UTC (1271026848)

- **Strain data used:**
  - K1:DCS-CALIB_STRAIN_C10
  - K1:DAC-STRAIN_C20

- **PEM channel used:**
  - K1:PEM-MIC_PSL_TABLE_PSL4_Z_OUT_DQ

- **Frequency Filter:** 55Hz ~ 65 Hz

- **Training Time:** 1270299936 ~ 1270300064 (128 second)

- **Cleaning Time:** 1270300064 ~ 1270300192 (128 second)
Denoising of 58.68Hz using Microphone: C10 (Low-latency)
Denoising of 58.68Hz using Microphone: C10 (Low-latency)
Denoising of 58.68Hz using Microphone: C10 (Low-latency)
Denoising of 58.68Hz using Microphone: C10 (Low-latency)
Denoising of 58.68Hz using Microphone: C20
Denoising of 58.68Hz using Microphone: C20
Denoising of 58.68Hz using Microphone: C20