https://iopscience.iop.org/article/10.1088/1361-6382/ad2194

https://dcc.ligo.org/P2500368

Building GSpyNetTree

A signal-vs-glitch classifier for gravitational-wave event candidates

ICERM SciML for GW astronomy @ Brown - Day 5

Sofía Álvarez-López, Mervyn Chan, Franz Herbst, Dhatri Raghunathan, Airene Ahuja, Annudesh Liyanage, Julian Ding, Raymond Ng, Jess McIver.

Gravitational-waves are really cool, and we can do a lot

of great science with them!

- But we need to make sure our detector data is clean to get the most science out of gravitational-waves.
- **Glitches** (short duration non-Gaussian noise) are problematic:

Subtracting glitches from gravitational-wave detector data during the third observing run

D Davis[1], T B Littenberg[2], I M Romero-Shaw[3,4,5], M Millhouse[6], J McIver[7], F Di Renzo[8,9], G Ashton[10]

Modeling and Reduction of High Frequency Scatter Noise at LIGO Livingston.

Siddharth Soni[1], Jane Glanzer[2], Anamaria Effler[3], Valera Frolov[3], Gabriela González[2], Arnaud Pele[4], Robert Schofield[5]

LIGO Detector Characterization in the Second and Third Observing Runs

D Davis[1], J S Areeda[2], B K Berger[3], R Bruntz[4], A Effler[5], R C Essick[6], R P Fisher[4], P Godwin[7], E Goetz[8,9,10],

Simulating Transient Noise Bursts in LIGO with Generative Adversarial Networks

Melissa Lopez,[1,2,a] Vincent Boudart,[3] Kerwin Buijsman,[4,5,b] Amit Reza,[1,2] and Sarah Caudill[1,2]

B. Berger, "Identification and mitigation of advanced LIGO noise sources", Journal of Physics: Conference Series 957, 012004 (2018).
D. Davis et al., "LIGO Detector Characterization in the Second and Third Observing Runs", Classical and Quantum Gravity 38, arXiv:2101.11673 [astro-ph, physics:gr-qc], 135014 (2021).
B. P. Abbott et al., "Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO's first observing run", Classical and Quantum Gravity 35, 065010, 065010 (2018).
J. Powell, "Parameter estimation and model selection of gravitational wave signals contaminated by transient detector noise glitches", Classical and Quantum Gravity 35, 155017 (2018).

Gravitational-waves are really cool, and we can do a lot of great science with them!

- But we need to make sure our detector data is clean to get the most science out of gravitational-waves.
- **Glitches** (short duration non-Gaussian noise) are problematic:

Generate false-positive candidates.

Corrupt data.

Bias astrophysical parameter estimation.

B. Berger, “Identification and mitigation of advanced LIGO noise sources”, Journal of Physics: Conference Series 957, 012004 (2018).
D. Davis et al., “LIGO Detector Characterization in the Second and Third Observing Runs”, Classical and Quantum Gravity 38, arXiv:2101.11673 [astro-ph, physics:gr-qc], 135014 (2021).
B. P. Abbott et al., “Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO’s first observing run”, Classical and Quantum Gravity 35, 065010, 065010 (2018).
J. Powell, “Parameter estimation and model selection of gravitational wave signals contaminated by transient detector noise glitches”, Classical and Quantum Gravity 35, 155017 (2018).

Gravitational-waves are really cool, and we can do a lot of great science with them!

- But we need to make sure our detector data is clean to get the most science out of gravitational-waves.
- **Glitches** (short duration non-Gaussian noise) pollute our data.

How do we tell apart a glitch from a gravitational-wave in a timely (and hopefully automated) fashion?

GSpyNetTree is a **signal**-vs-**glitch** classifier for gravitational-wave event candidates.

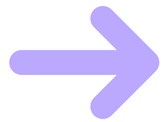
Running as a LIGO-Virgo-KAGRA event validation tool since ER15 (right before O4 started)!

Where do we need a signal-vs-glitch classifier?

A new GW candidate
is detected

Where do we need a signal-vs-glitch classifier?

A new GW candidate is detected



Event is stored in GraceDB

Where do we need a signal-vs-glitch classifier?

A new GW candidate is detected

GraceDB

Event is stored in GraceDB

Data Quality Report

Collection of tools used to enable/support the vetting of non-astrophysical candidates.

Where do we need a signal-vs-glitch classifier?

A new GW candidate is detected

GraceDB

Event is stored in GraceDB

Data Quality Report

Collection of tools used to enable/support the vetting of non-astrophysical candidates.

iDQ

. . .

GSpyNetTree

. . .

Glitch Find

Where do we need a signal-vs-glitch classifier?

A new GW candidate is detected

GraceDB

Event is stored in GraceDB

Data Quality Report

Collection of tools used to enable/support the vetting of non-astrophysical candidates.

iDQ

. . .

GSpyNetTree

. . .

Glitch Find

Decide whether event is astrophysical or not.

Where do we need a signal-vs-glitch classifier?

In our initial pipeline! To tell whether candidate event is (or not) a glitch.

A new GW candidate is detected

GraceDB

Event is stored in GraceDB

Data Quality Report

Collection of tools used to enable/support the vetting of non-astrophysical candidates.

iDQ

. . .

GSpyNetTree

. . .

Glitch Find

Decide whether event is astrophysical or not.

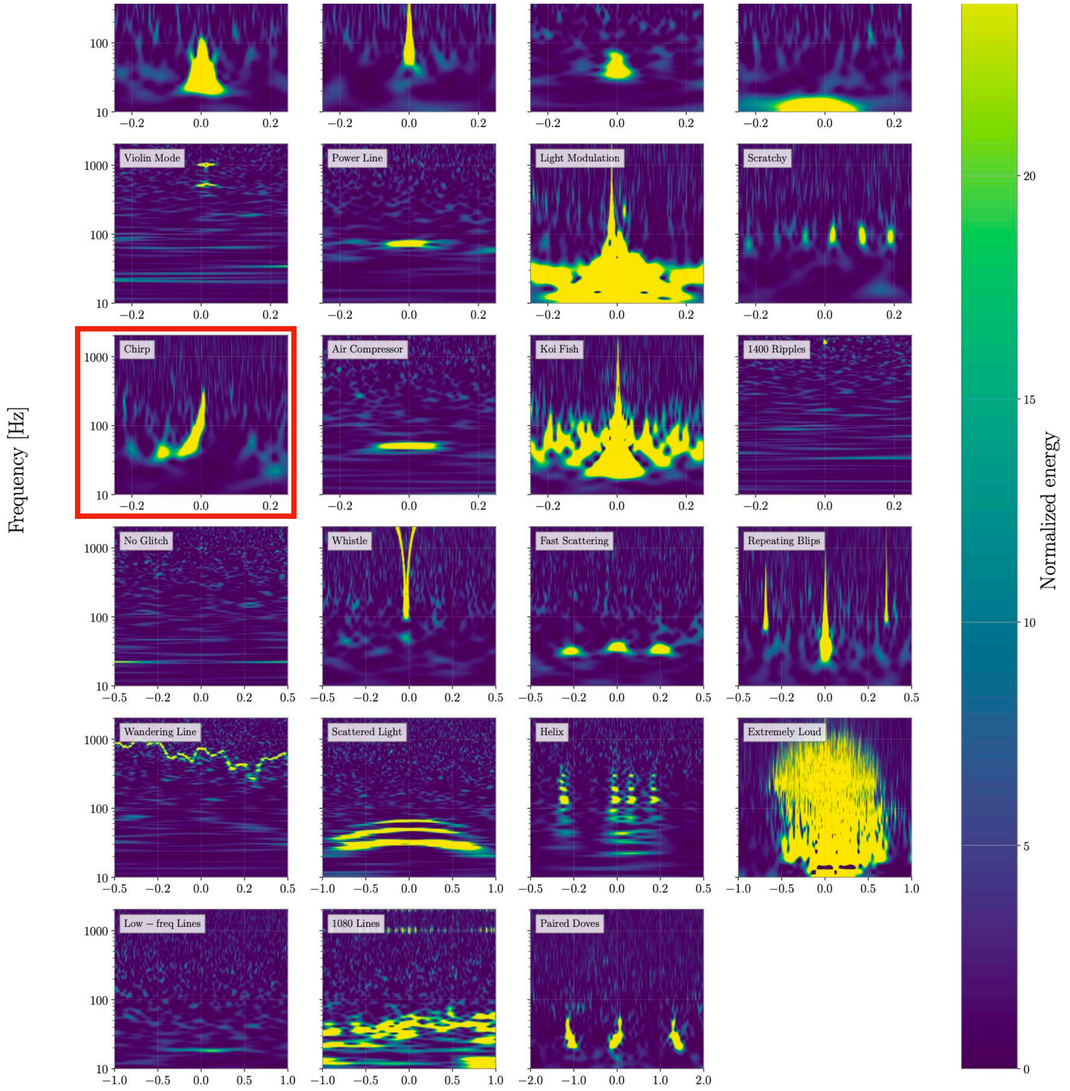
What considerations should we take into account to build a signal-vs-glitch classifier?

1. Glitches come in various time-frequency morphologies.

1. Glitches come in various time-frequency morphologies.

We saw this in **Gravity Spy!**

A MULTI-CLASS Convolutional Neural Network (CNN) image classifier



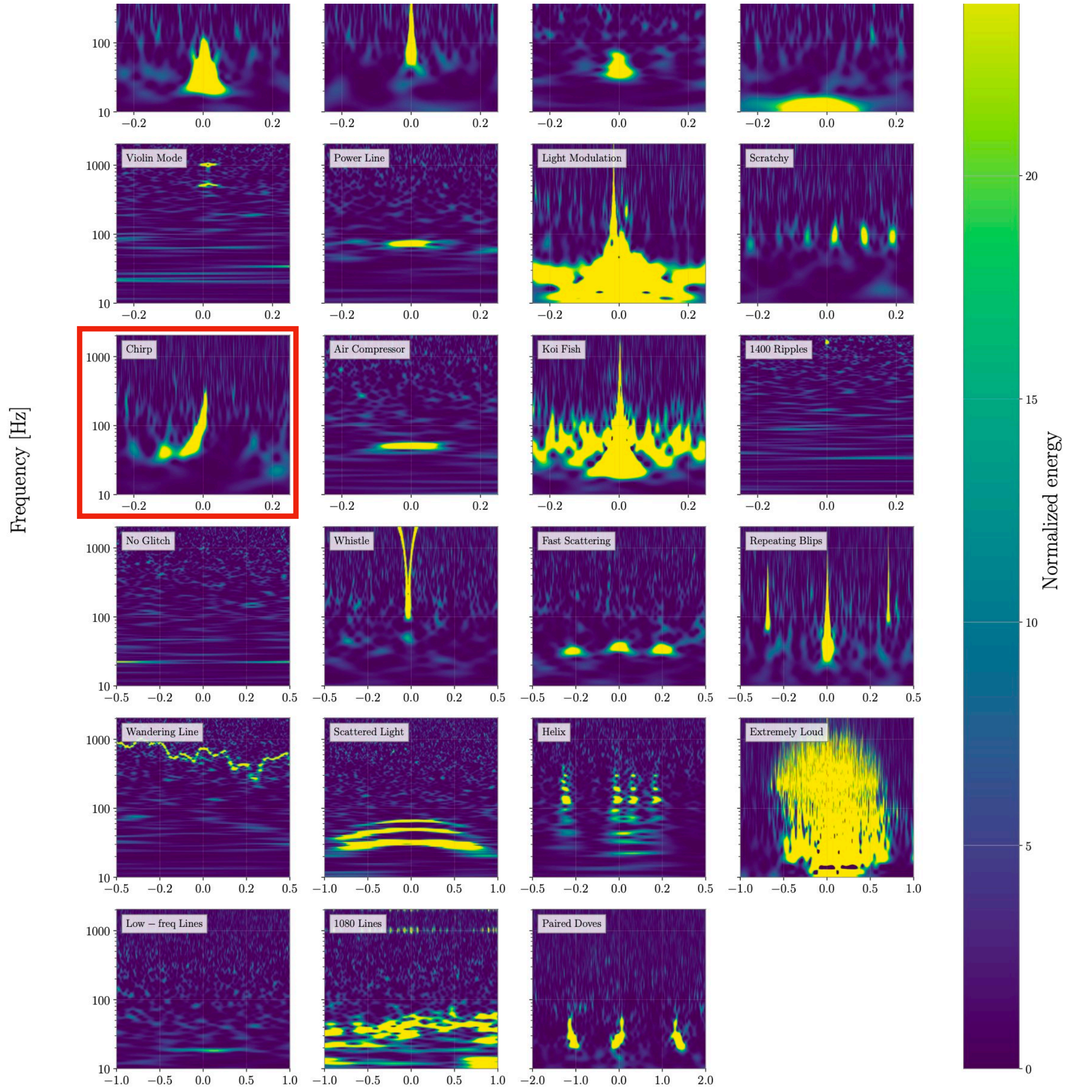
M. Zevin et al., “Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science”, Classical and Quantum Gravity 34, arXiv:1611.04596 [astro-ph, physics:grqc, physics:physics], 064003 (2017).

1. Glitches come in various time-frequency morphologies.

We saw this in **Gravity Spy!**

A MULTI-CLASS Convolutional Neural Network (CNN) image classifier

- CNNs: Machine Learning architecture used for image classification tasks.
- MULTI-CLASS: Distinguishes 20 different glitch classes and 1 Chirp class (red box) via spectrograms.
- Successful glitch classification in previous observing runs.



M. Zevin et al., “Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science”, Classical and Quantum Gravity 34, arXiv:1611.04596 [astro-ph, physics:grqc, physics:physics], 064003 (2017).

1. Glitches come in various time-frequency morphologies.

They are particularly challenging when they mimic the form of GW events.

M. Zevin et al., “Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science”, Classical and Quantum Gravity 34, arXiv:1611.04596 [astro-ph, physics:grqc, physics:physics], 064003 (2017).

1. Glitches come in various time-frequency morphologies.

They are particularly challenging when they mimic the form of GW events.

Potential to use Gravity Spy's architecture to build

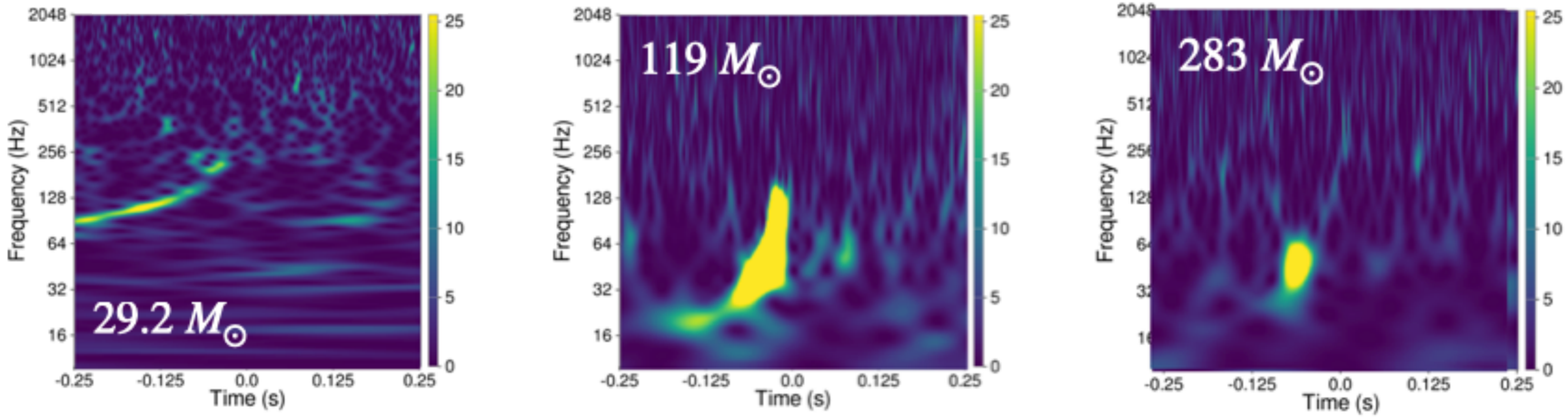
GSpyNetTree

Gravity Spy Convolutional Neural Network Decision Tree

M. Zevin et al., "Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science", Classical and Quantum Gravity 34, arXiv:1611.04596 [astro-ph, physics:grqc, physics:physics], 064003 (2017).

Why a decision tree?

Important effect of candidate mass on GW morphology in time-frequency visualizations.

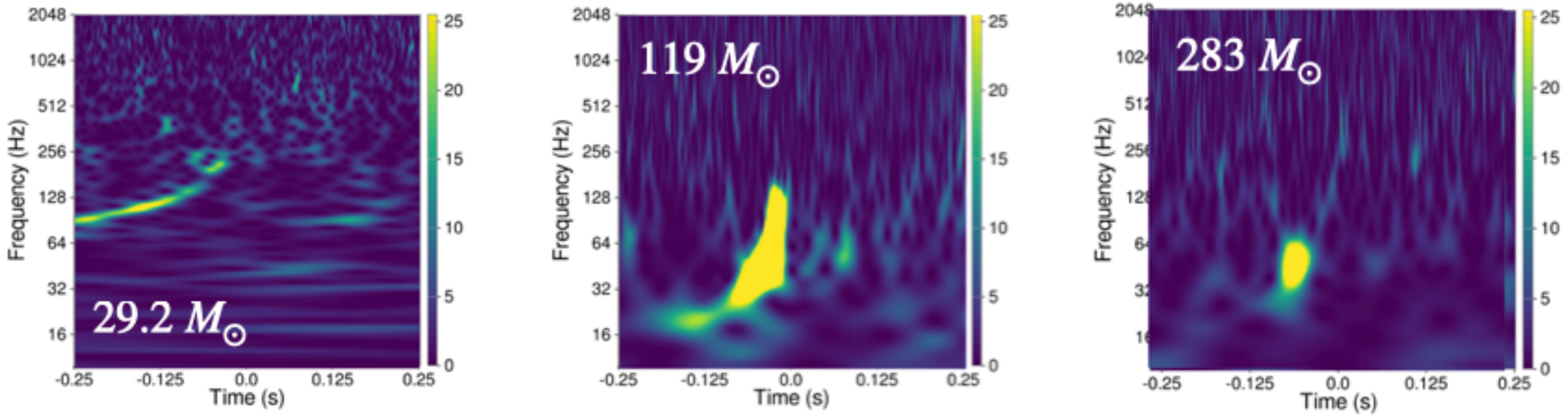


Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

Why a decision tree?

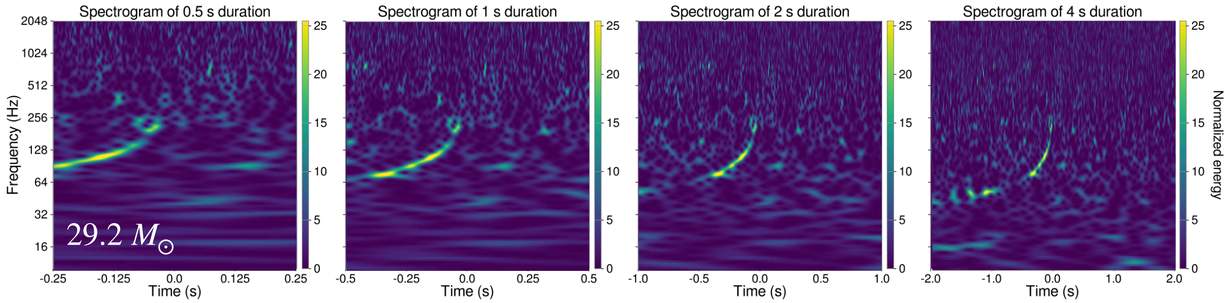
Important effect of candidate mass on GW morphology in time-frequency visualizations.

Depending on their total mass, some candidates are more morphologically similar to certain types of glitches.

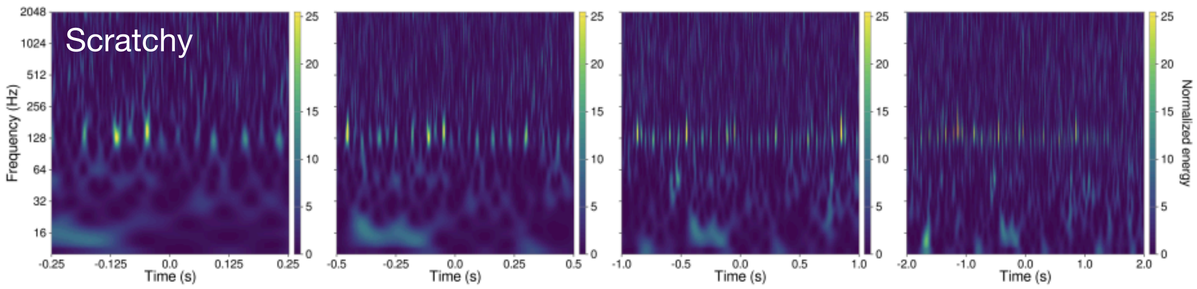


Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

GW samples

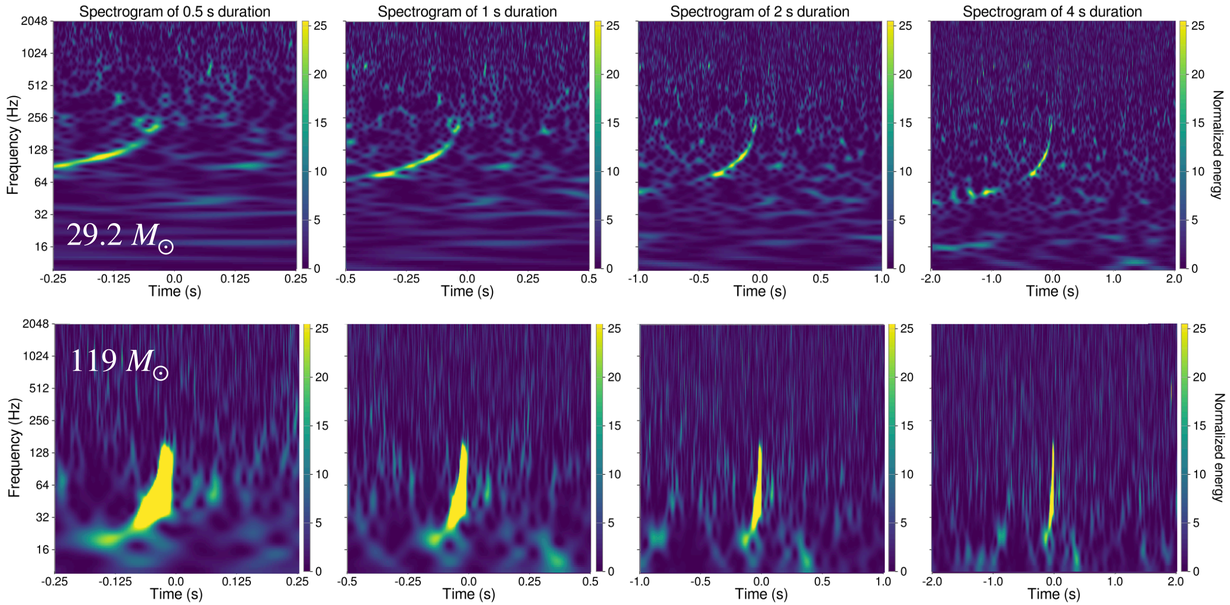


Glitch samples

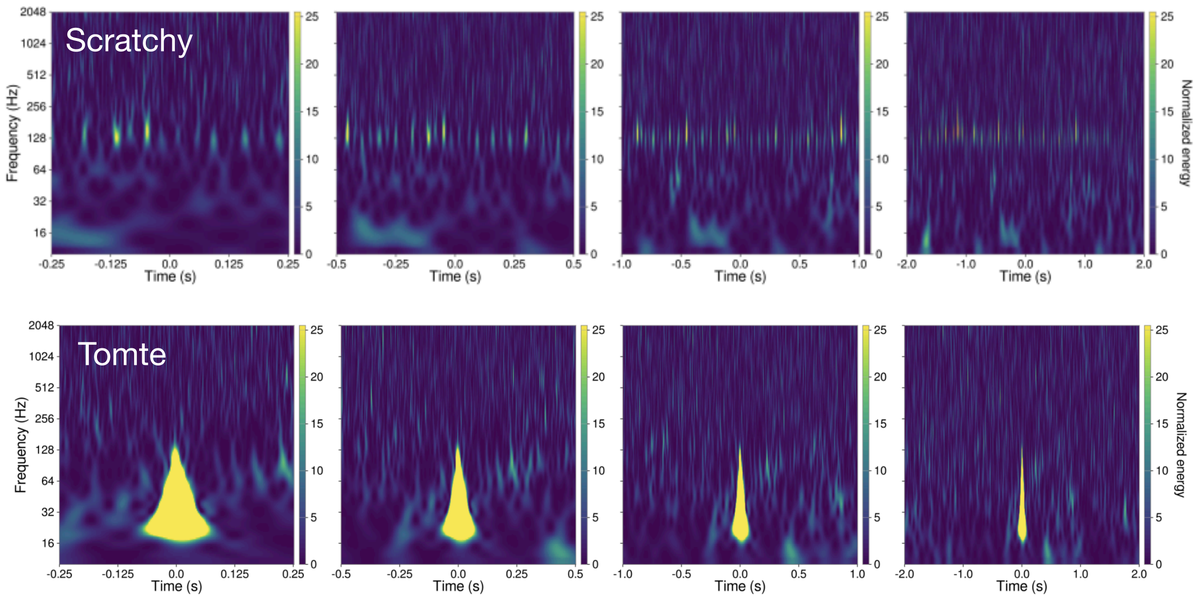


Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

GW samples

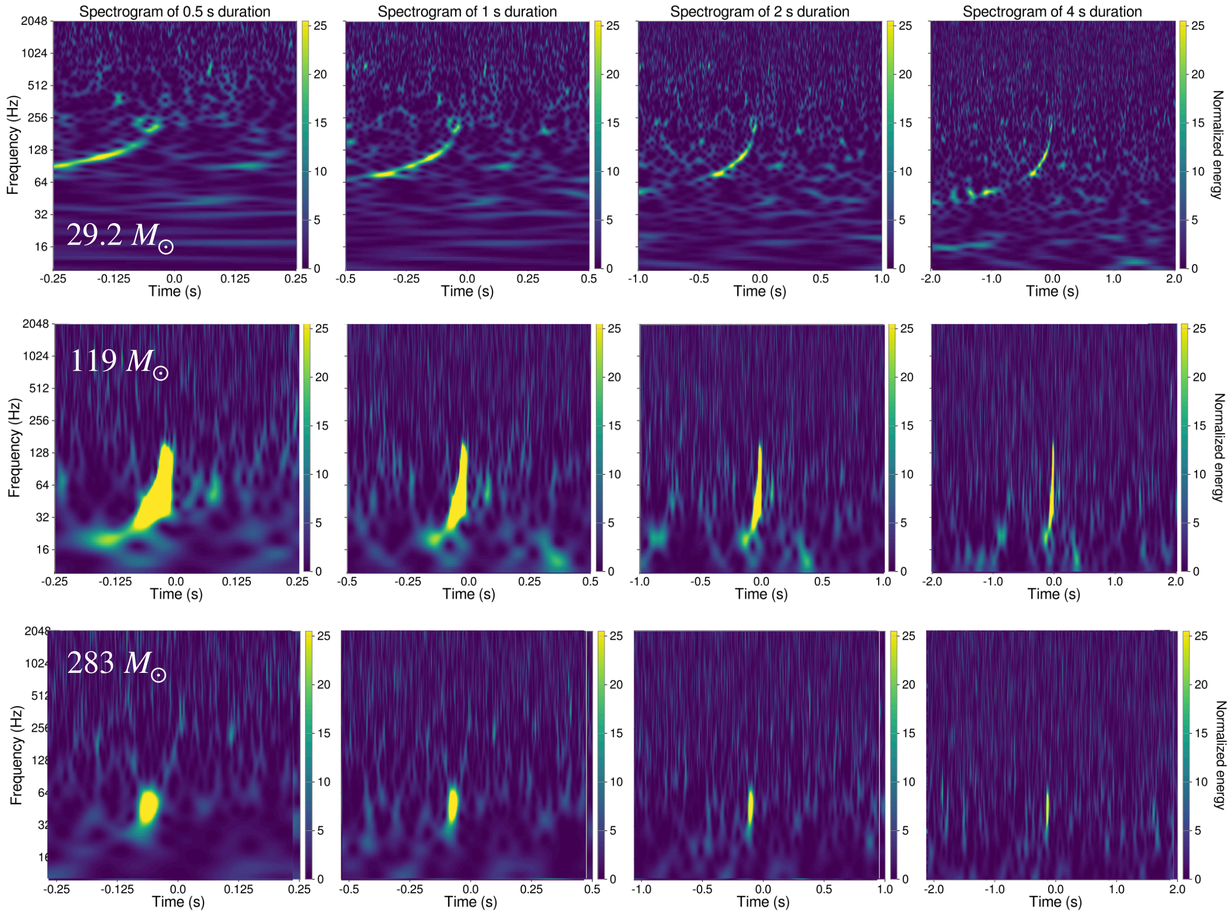


Glitch samples

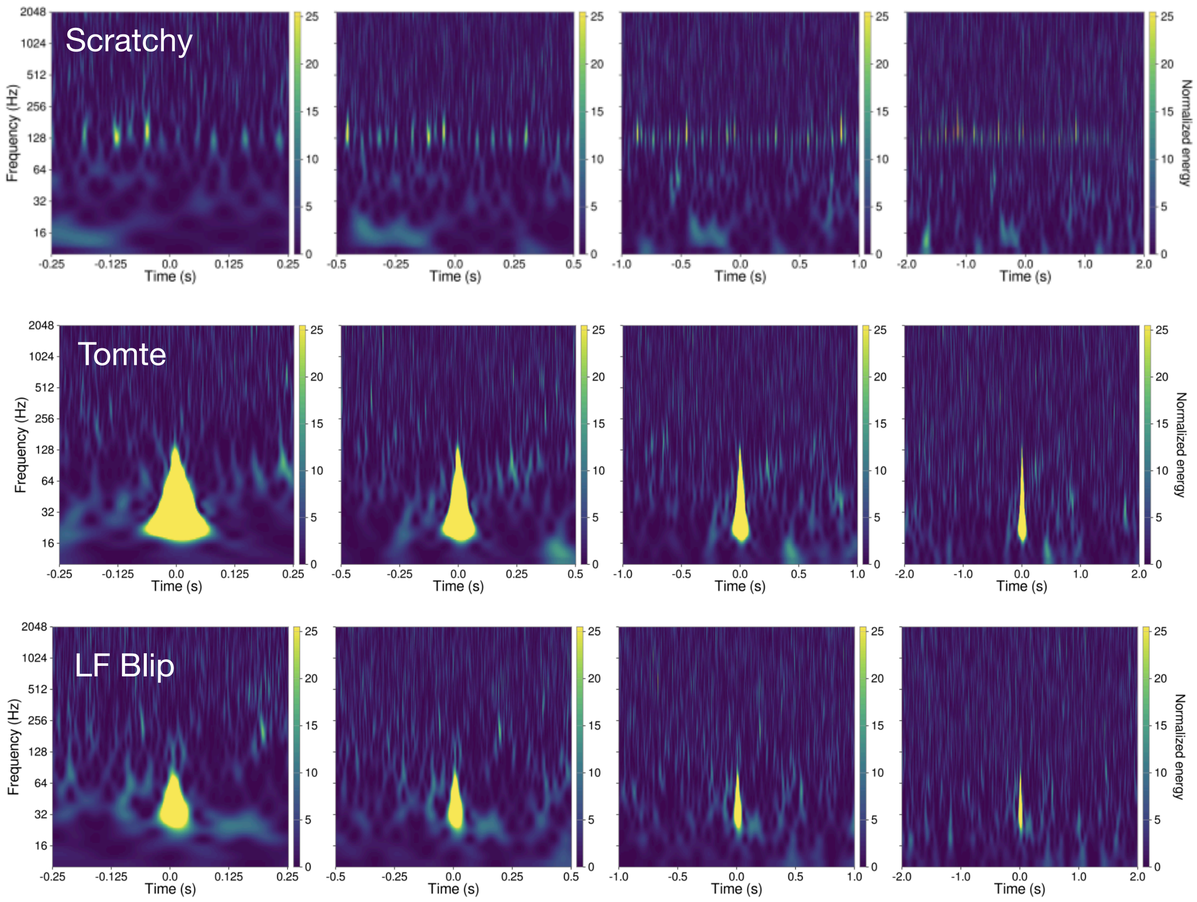


Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

GW samples

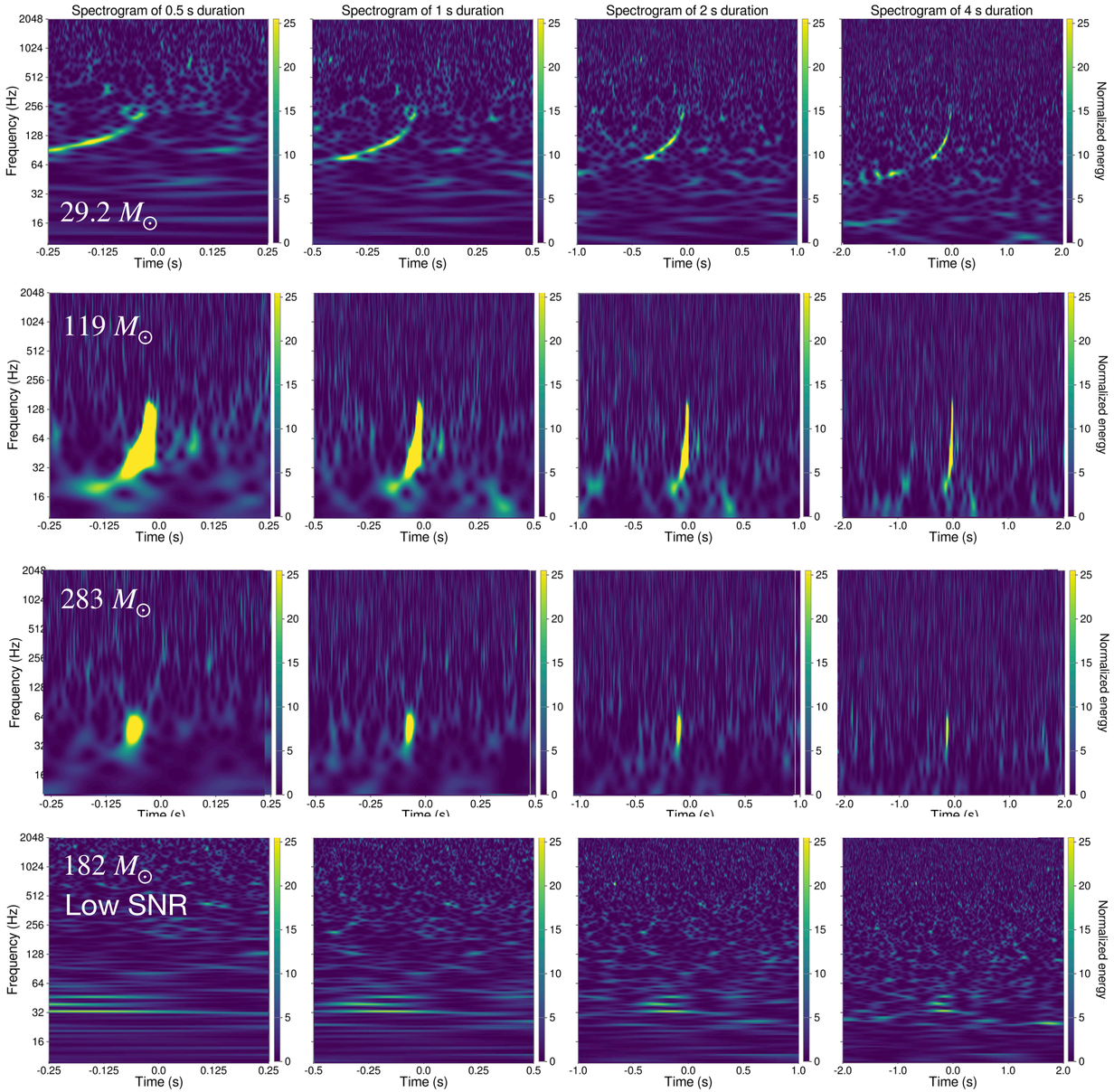


Glitch samples

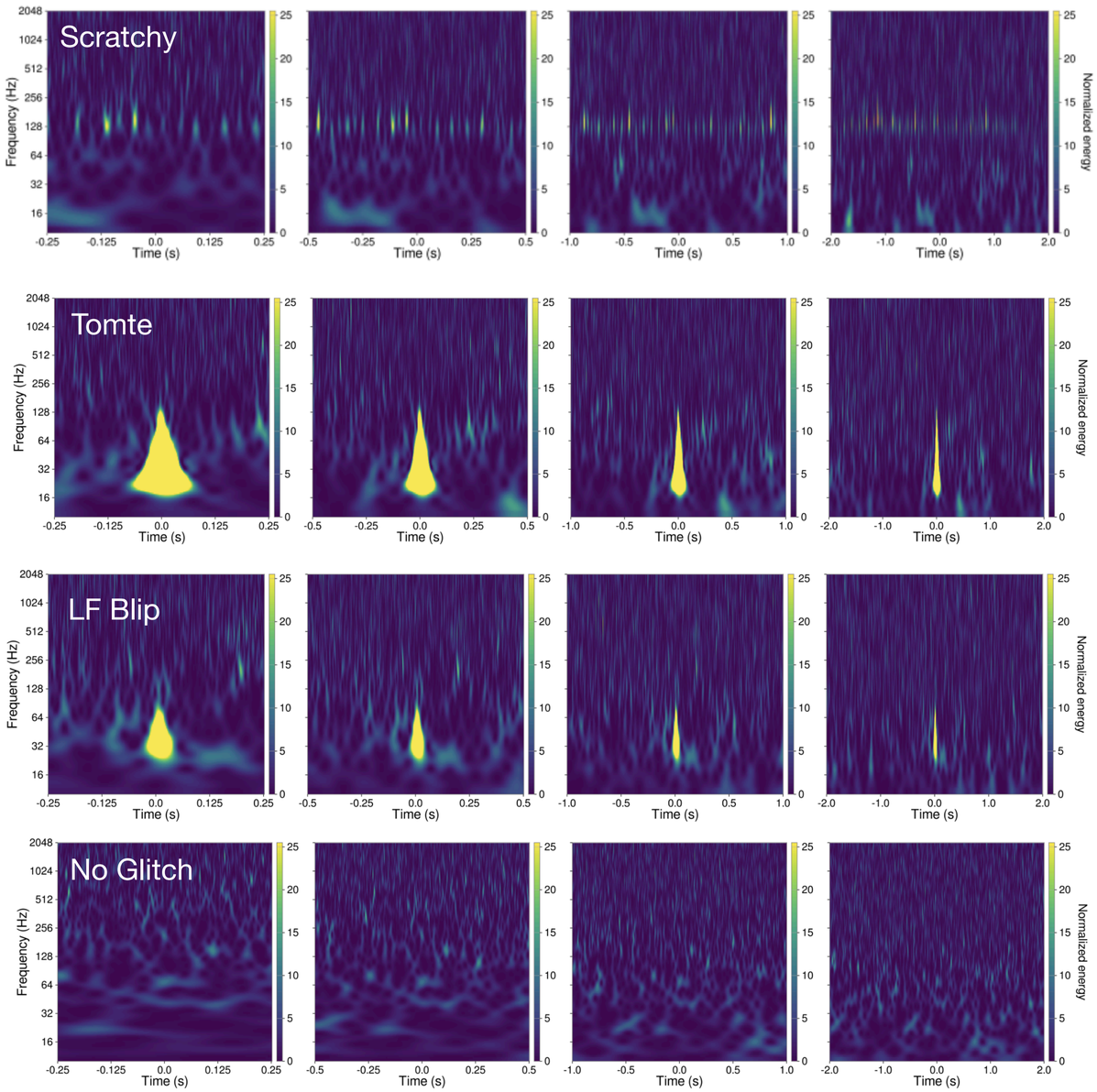


Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

GW samples



Glitch samples



Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

GSpyNetTree's philosophy is:

Different morphologies, separate classifiers

GSpyNetTree considers three classifiers, depending on the total mass of the candidate.

Low-mass classifier
$5 - 50\ M_\odot$

High-mass classifier
$50 - 250\ M_\odot$

Extremely high-mass classifier
$> 250\ M_\odot$

Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

D. Davis et al., "Subtracting glitches from gravitational-wave detector data during the third LIGO-virgo observing run", Classical and Quantum Gravity 39, 245013 (2022).

GSpyNetTree's philosophy is:

Different morphologies, separate classifiers

GSpyNetTree considers three classifiers along with **morphologically similar glitches.**

Low-mass classifier
$5 - 50\ M_\odot$

1. Blip
2. Low-frequency Blip
3. Koi Fish
4. Scratchy

High-mass classifier
$50 - 250\ M_\odot$

1. Blip
2. Low-frequency Blip
3. Koi Fish
4. Tomte

Extremely high-mass classifier
$> 250\ M_\odot$

1. Blip
2. Low-frequency Blip

Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

D. Davis et al., "Subtracting glitches from gravitational-wave detector data during the third LIGO-virgo observing run", Classical and Quantum Gravity 39, 245013 (2022).

GSpyNetTree's philosophy is:

Different morphologies, separate classifiers

GSpyNetTree considers three classifiers along with **morphologically similar glitches.** We also include **No Glitch** samples to account for low SNR GWs.

Low-mass classifier
$5 - 50\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**
4. Koi Fish
5. Scratchy

High-mass classifier
$50 - 250\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**
4. Koi Fish
5. Tomte

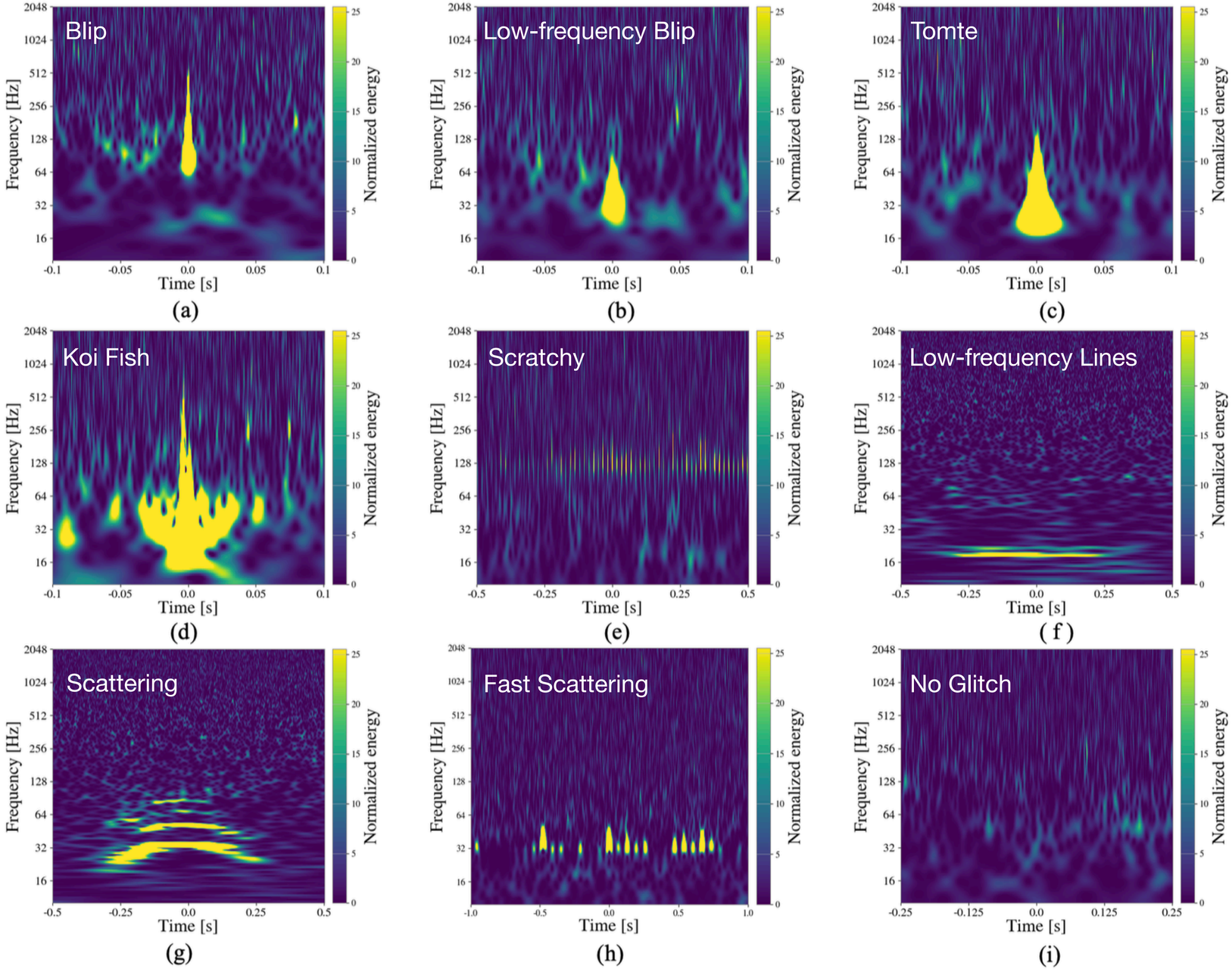
Extremely high-mass classifier
$> 250\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**

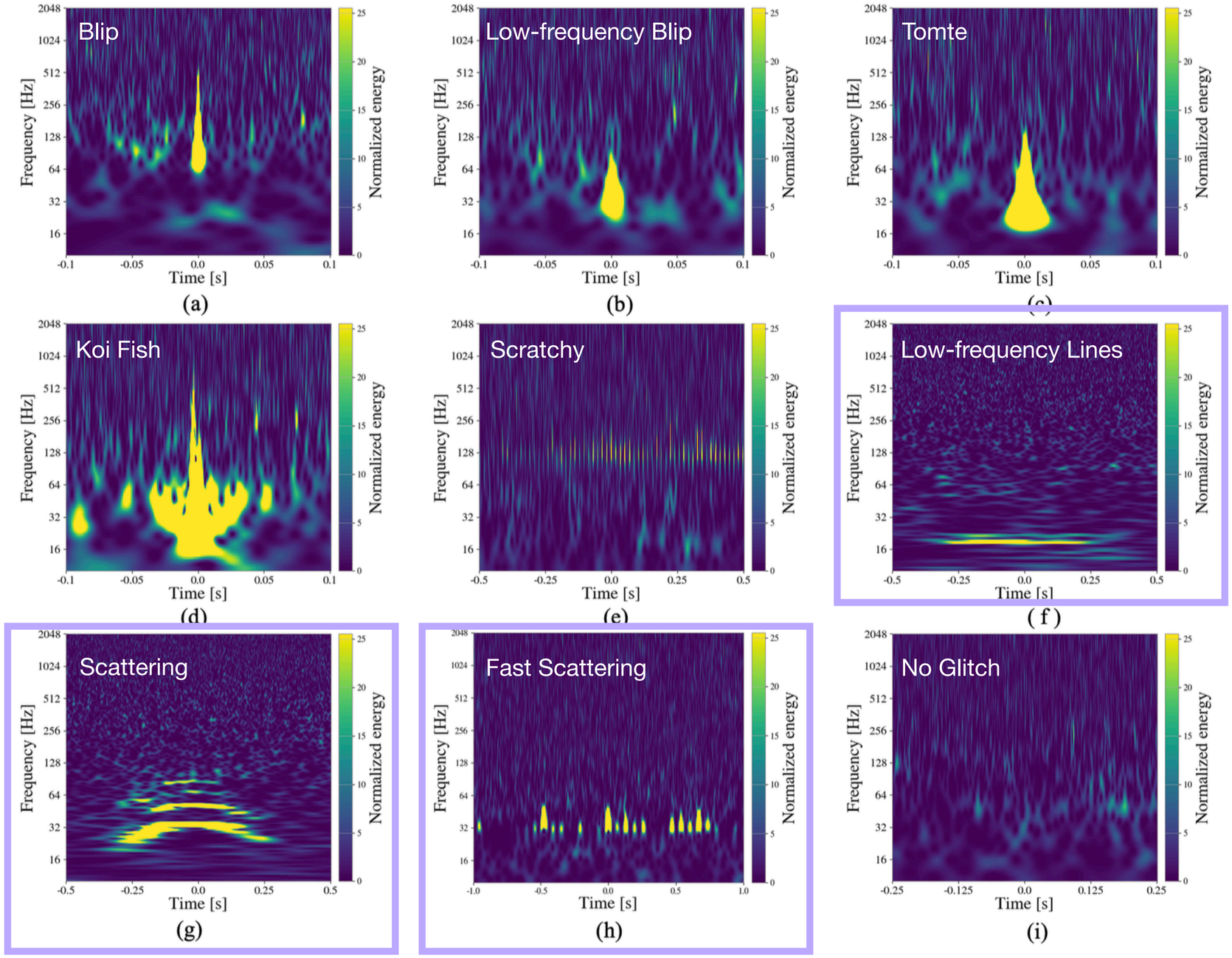
Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

D. Davis et al., "Subtracting glitches from gravitational-wave detector data during the third LIGO-virgo observing run", Classical and Quantum Gravity 39, 245013 (2022).

[13] Álvarez-López et al., “GSpyNetTree: A signal-vs-glitch classifier for gravitational-wave event candidates”, (

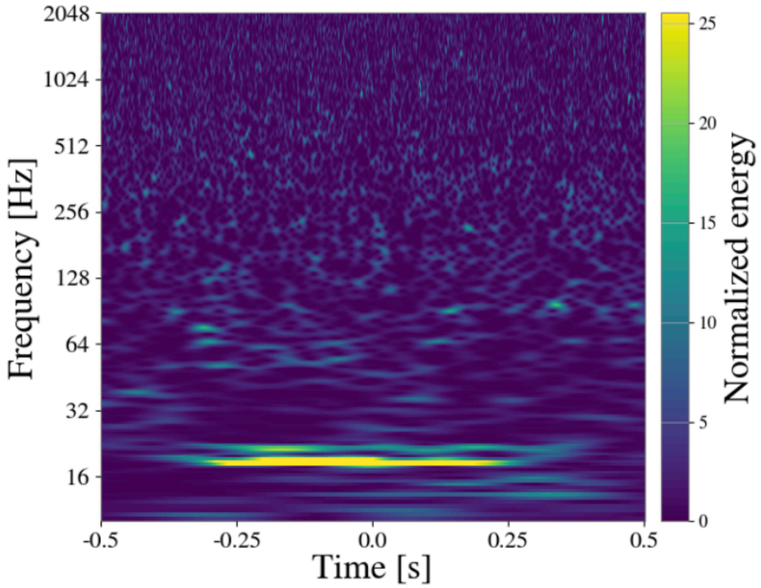


[13] Álvarez-López et al., “GSpyNetTree: A signal-vs-glitch classifier for gravitational-wave event candidates”,

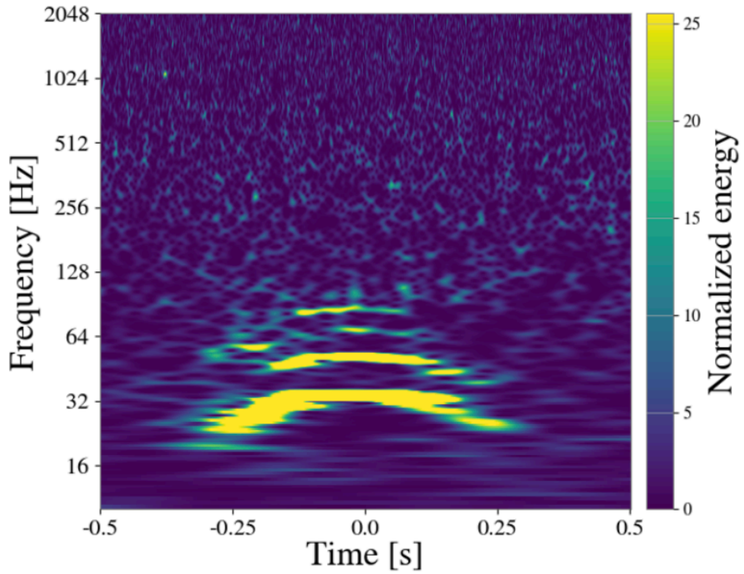


Why include these three types of glitches in all classifiers?

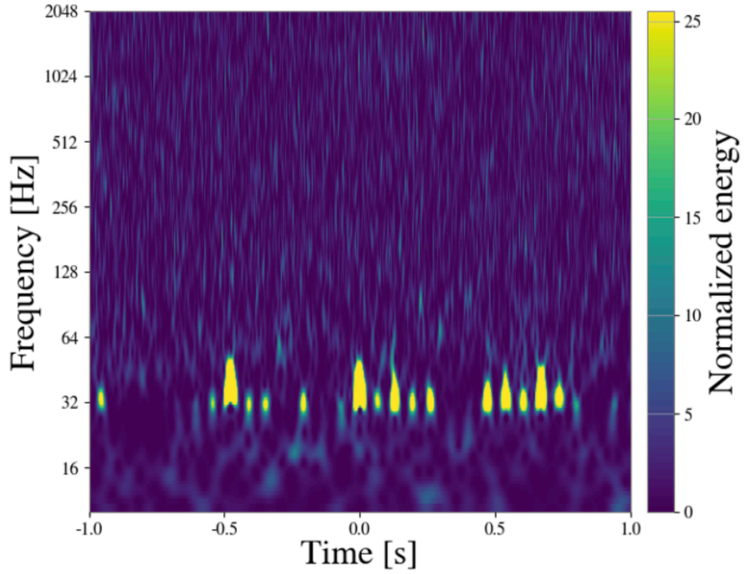
Low-frequency Lines



Scattering



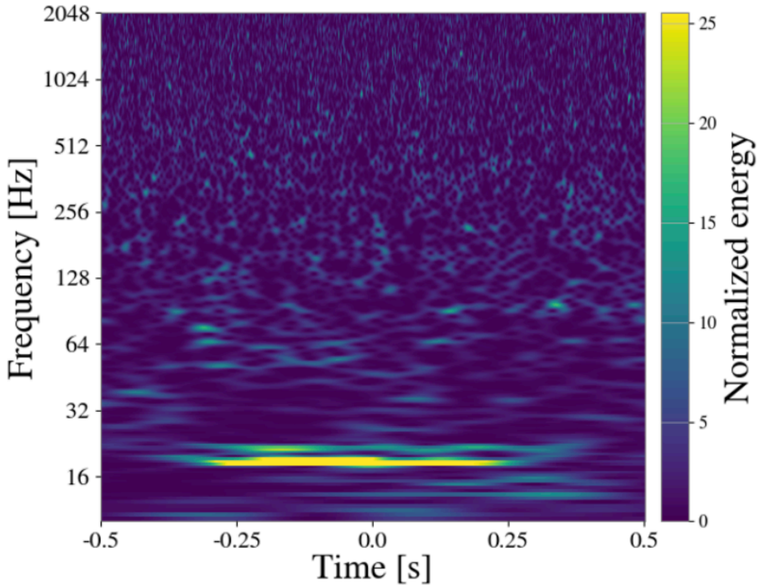
Fast Scattering



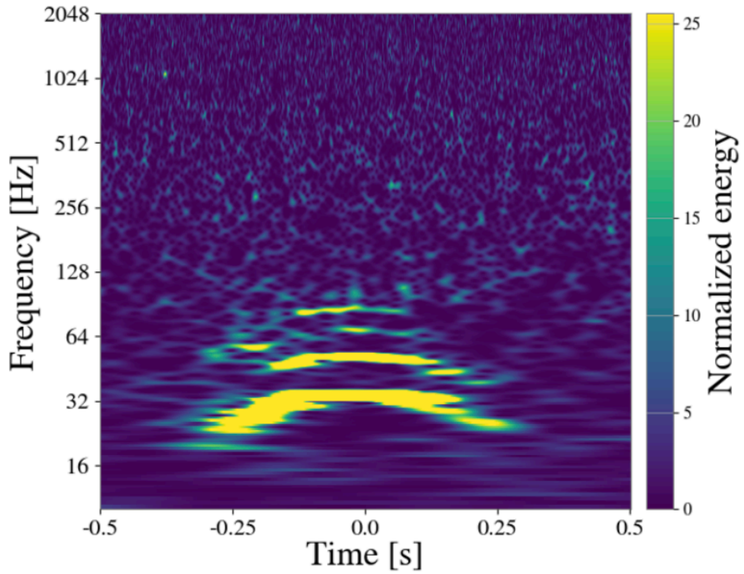
Why include these three types of glitches in all classifiers?

Because they are the most common!

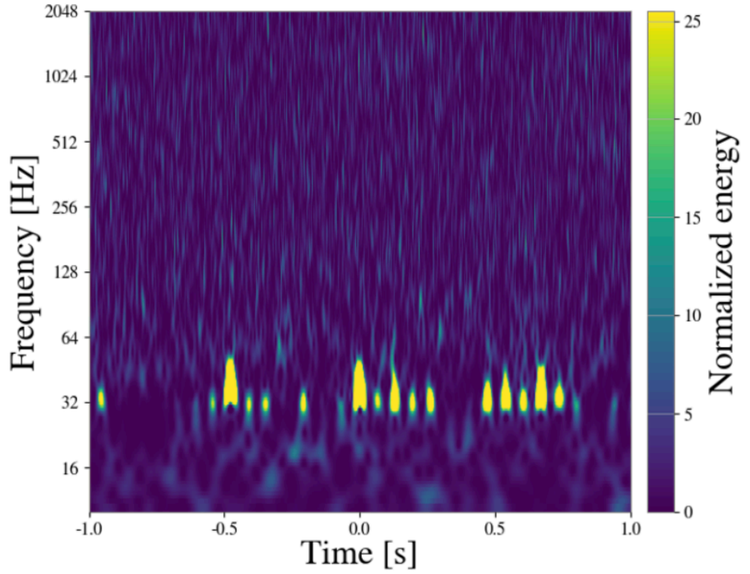
Low-frequency Lines



Scattering



Fast Scattering



Low-frequency noise was very common during ER15.

Most common glitches in O3, responsible for broad ranges of vetted times.

GSpyNetTree's philosophy is:

Different morphologies, separate classifiers

+ most common glitches!

GSpyNetTree considers three classifiers along with morphologically similar glitches and most common detector glitches. We also include **No Glitch** samples to account for low SNR GWs.

Low-mass classifier
$5 - 50\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**
4. Koi Fish
5. Scratchy
6. Low-frequency Lines
7. Scattering
8. Fast Scattering

High-mass classifier
$50 - 250\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**
4. Koi Fish
5. Tomte
6. Low-frequency Lines
7. Scattering
8. Fast Scattering

Extremely high-mass classifier
$> 250\ M_\odot$

1. Blip
2. Low-frequency Blip
3. **No Glitch**
4. Low-frequency Lines
5. Scattering
6. Fast Scattering

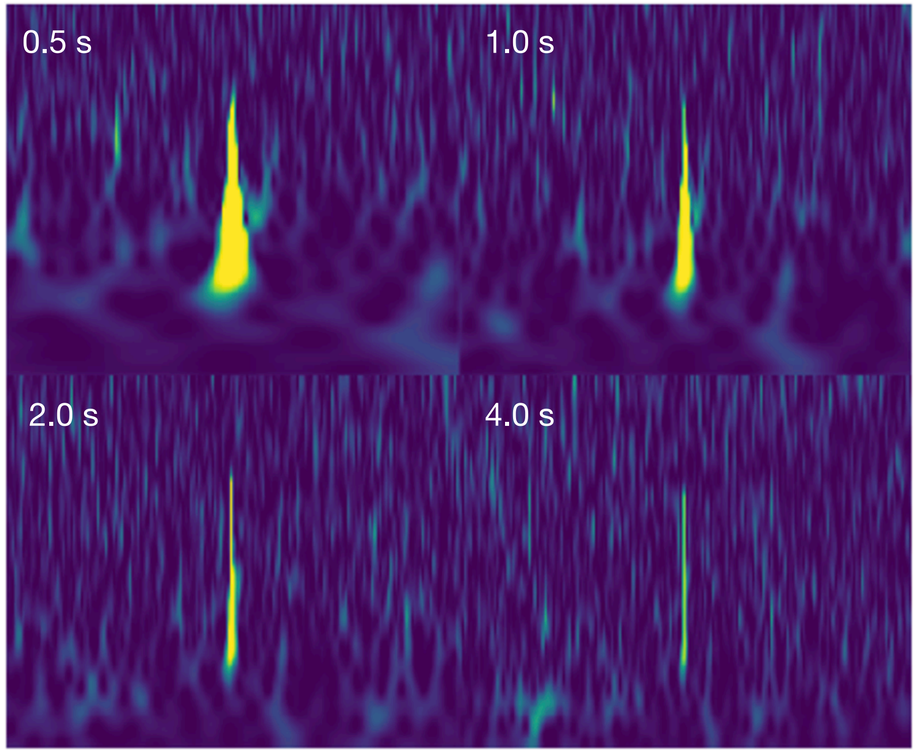
Jarov, S., Thiele, S., Soni, S., Ding, J., McIver, J., Ng, R., ... & Davis, D. (2023). A new method to distinguish gravitational-wave signals from detector noise transients with gravity spy. *arXiv preprint arXiv:2307.15867*.
Alvarez-Lopez, S., Liyanage, A., Ding, J., Ng, R., & McIver, J. (2024). GSpyNetTree: a signal-vs-glitch classifier for gravitational-wave event candidates. *Classical and Quantum Gravity*, *41*(8), 085007.

D. Davis et al., "Subtracting glitches from gravitational-wave detector data during the third LIGO-virgo observing run", Classical and Quantum Gravity 39, 245013 (2022).

How to account for shorter/longer morphologies?

Just like in Gravity Spy, we use a 2x2 matrix of spectrograms of different durations

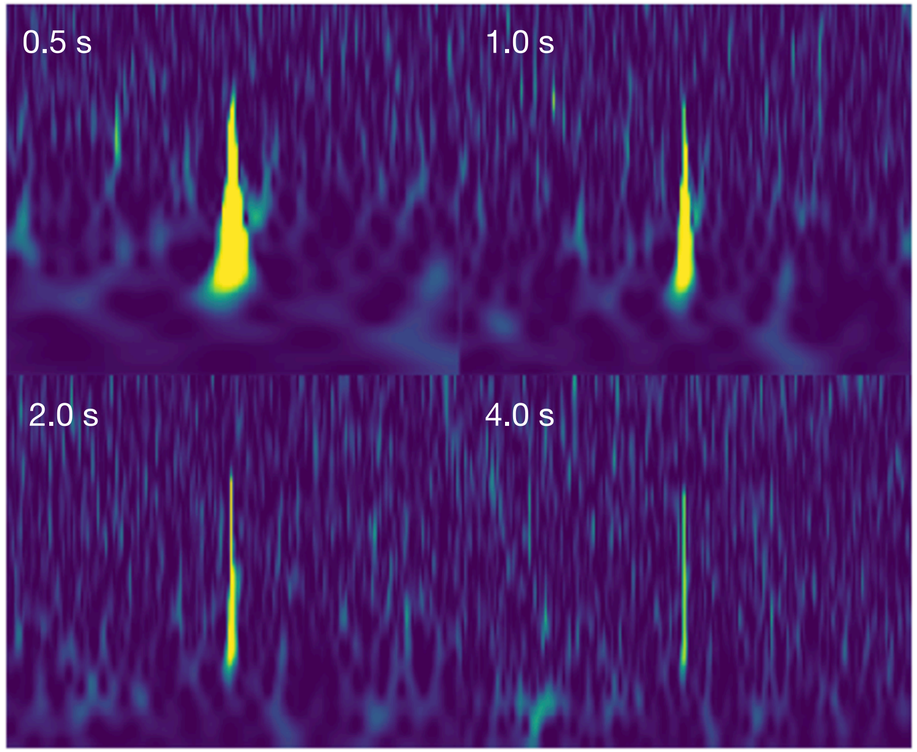
- 4 spectrograms, 0.5 s, 1 s, 2 s, and 4 s in duration arranged in a 2x2 matrix.



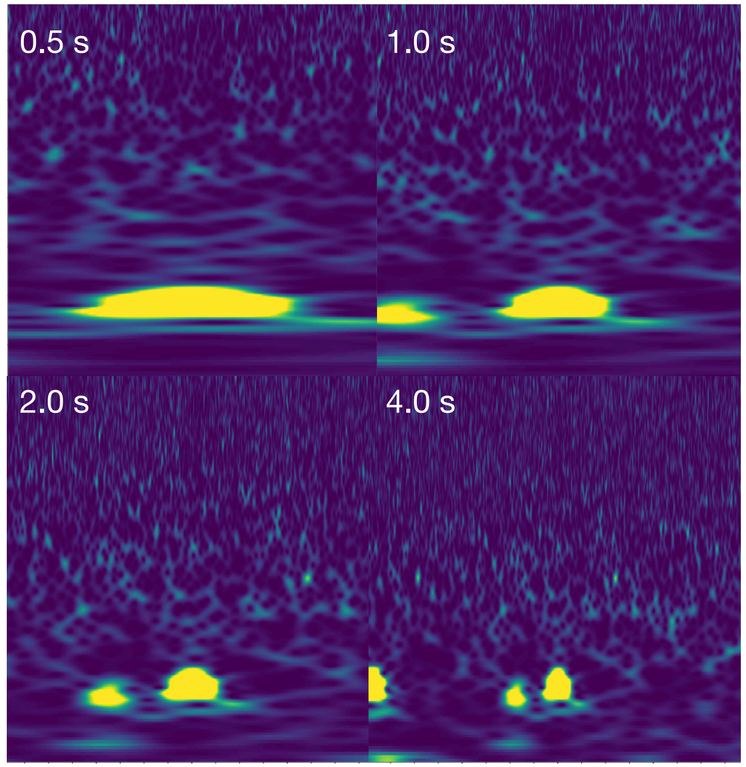
How to account for shorter/longer morphologies?

Just like in Gravity Spy, we use a 2x2 matrix of spectrograms of different durations

- 4 spectrograms, 0.5 s, 1 s, 2 s, and 4 s in duration arranged in a 2x2 matrix.



Blip



Fast Scattering

What considerations should we take into account when building a **signal-vs-glitch classifier?**

1. Glitches come in various time-frequency morphologies.
    - 1.1 Some glitches are more common than others.
        - 1.2 How to account for long/short morphologies?

What considerations should we take into account when building a **signal-vs-glitch classifier?**

1 • Glitches come in various time-frequency morphologies. ✓

Three separate classifiers sorted via candidate mass

1.1 Some glitches are more common than others. ✓

Include most common glitches during O3/ER15.

1.2 How to account for short/long morphologies? ✓

Consider spectrograms of 4 different durations.

? What if new glitch classes show up?

What considerations should we take into account when building a **signal-vs-glitch classifier?**

1. Glitches come in various time-frequency morphologies. ✓

Three separate classifiers sorted via candidate mass

1.1 Some glitches are more common than others. ✓

Include most common glitches during O3/ER15.

1.2 How to account for short/long morphologies? ✓

Consider spectrograms of 4 different durations.

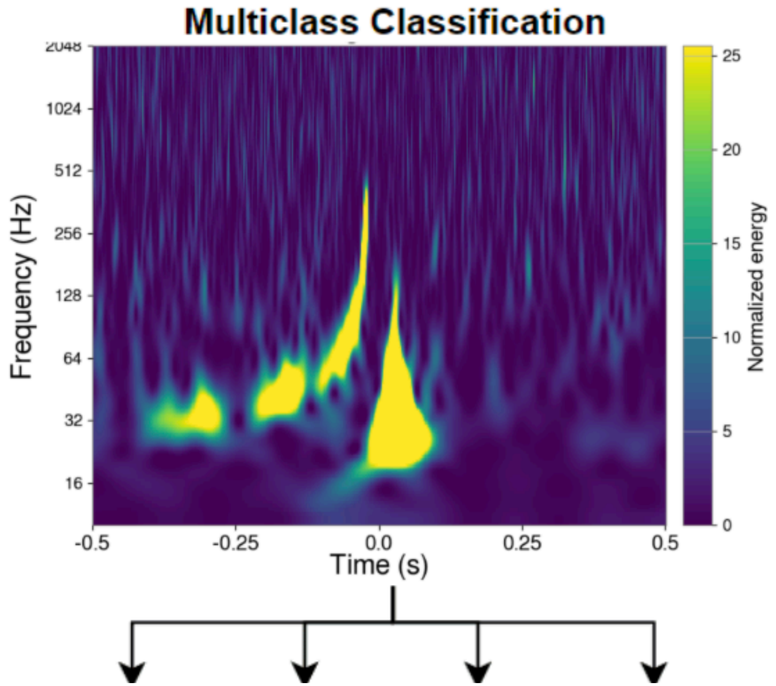
2. Glitch rate is about ~1/minute

2. Glitch rate is about ~1/minute

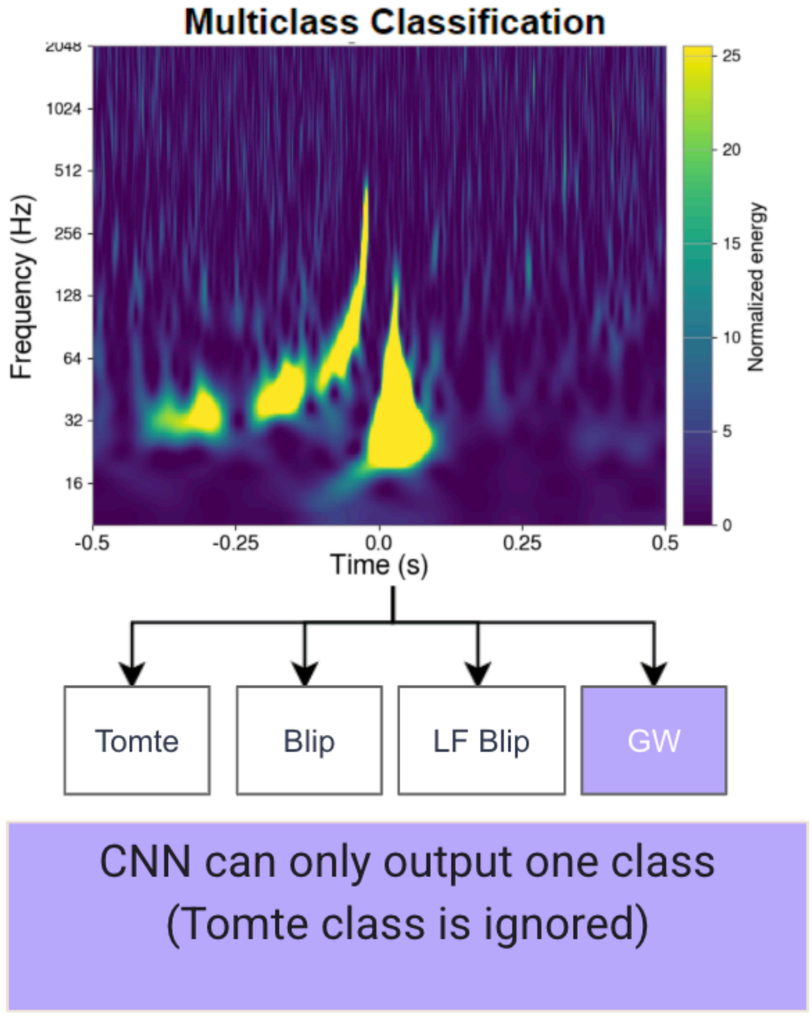
Probability that glitch overlaps/occurs in the close proximity of GW candidate is ~high.

- May lead to unnecessarily vetoed candidates.
- Needs to be addressed! Happened in **24% of O3 candidates**, and has happened in O4.

Is Gravity Spy's multi-class architecture sufficient?

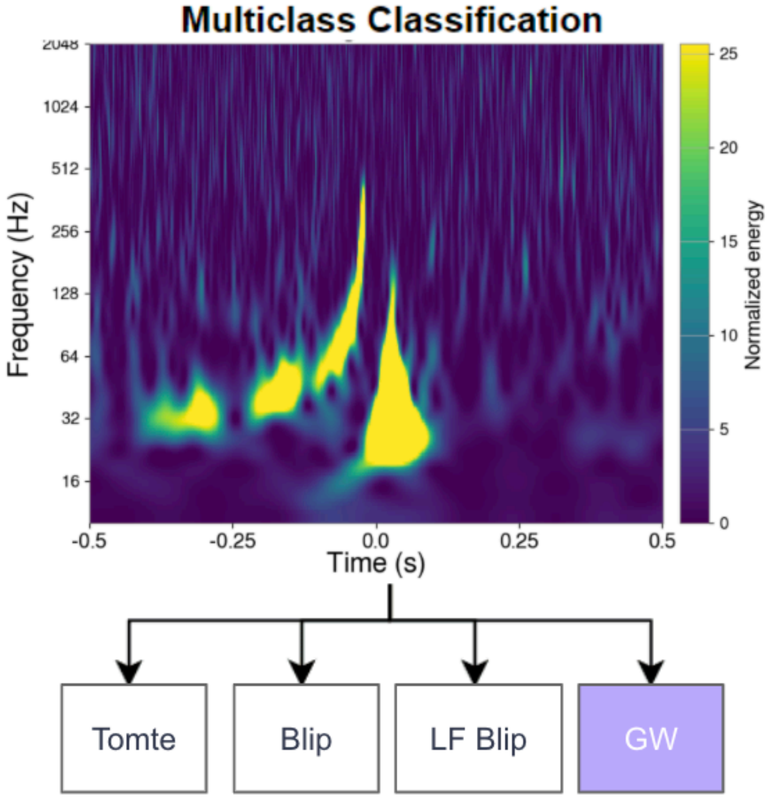


Is Gravity Spy's multi-class architecture sufficient?

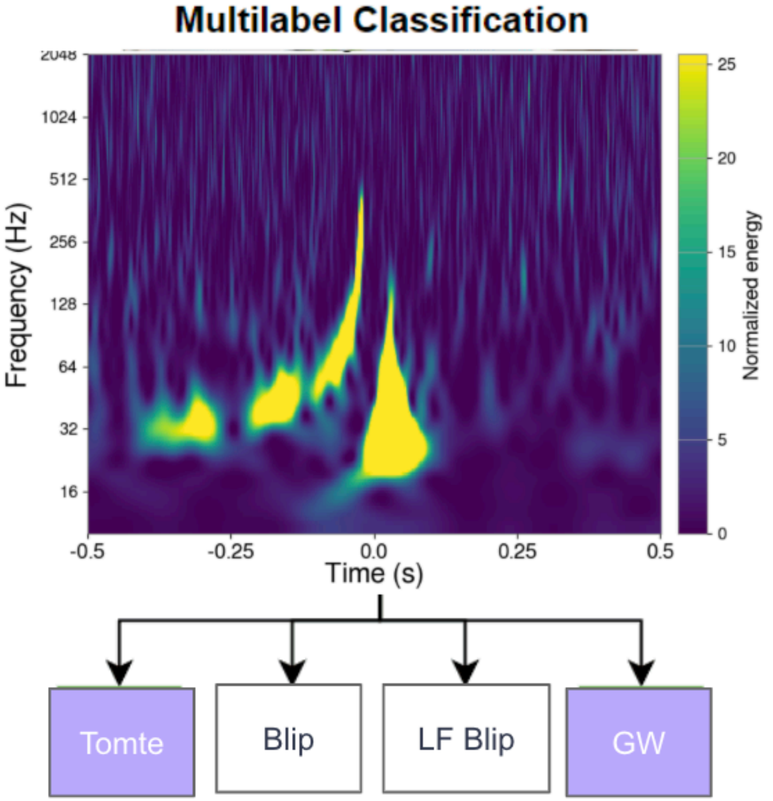


CNN can only output one class
(Tomte class is ignored)

Use a multilabel architecture instead!



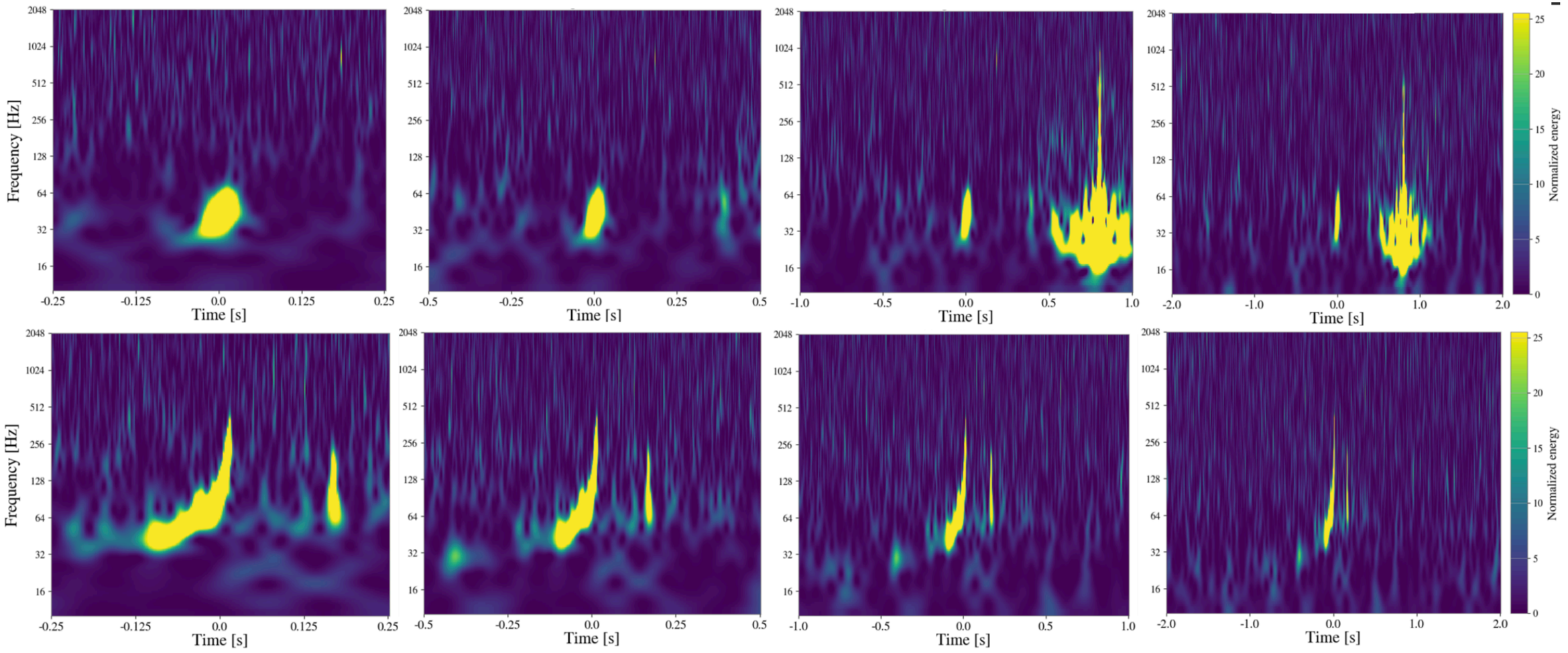
CNN can only output one class
(Tomte class is ignored)



CNN may predict 0+ classes.

Making GSpyNetTree robust to overlapping

samples of GWs and glitches



What considerations should we take into account when building a **signal-vs-glitch classifier?**

1. Glitches come in various time-frequency morphologies. ✓

Three separate classifiers sorted via candidate mass

1.1 Some glitches are more common than others. ✓

Include most common glitches during O3/ER15.

1.2 How to account for short/long morphologies? ✓

Consider spectrograms of 4 different durations.

2. Glitch rate is about ~1/minute

What considerations should we take into account when building a **signal-vs-glitch classifier?**

1. Glitches come in various time-frequency morphologies. ✓

Three separate classifiers sorted via candidate mass

1.1 Some glitches are more common than others. ✓

Include most common glitches during O3/ER15.

1.2 How to account for short/long morphologies? ✓

Consider spectrograms of 4 different durations.

2. Glitch rate is about ~1/minute ✓

3. Effect of (changes in) background noise in spectrogram classification.

3. Effect of (changes in) background noise in spectrogram classification.

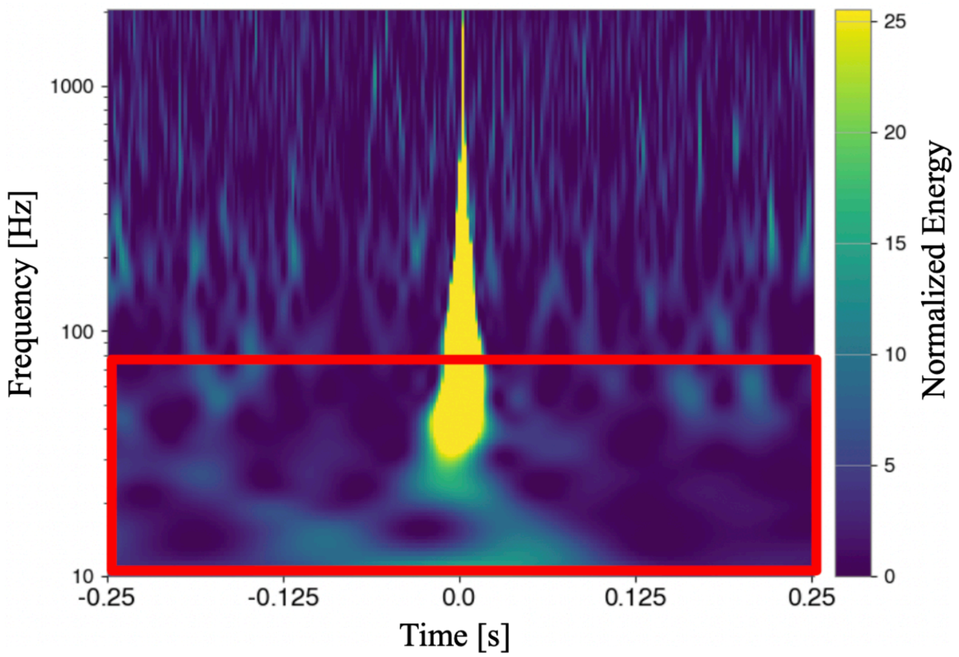
What is the problem?

- Background noise changes from one observing run to another (due to new calibrations in the detectors).

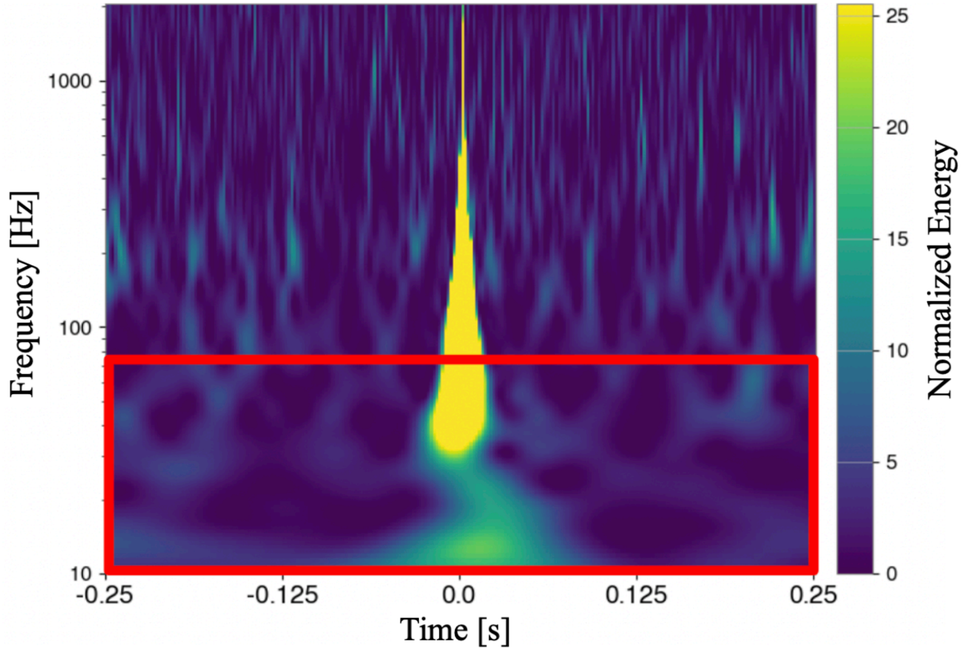
- Background noise also changes between interferometers.

We expected O4 low-latency data to apply a non-linear subtraction of 60 Hz power artifacts.

Subtraction applied

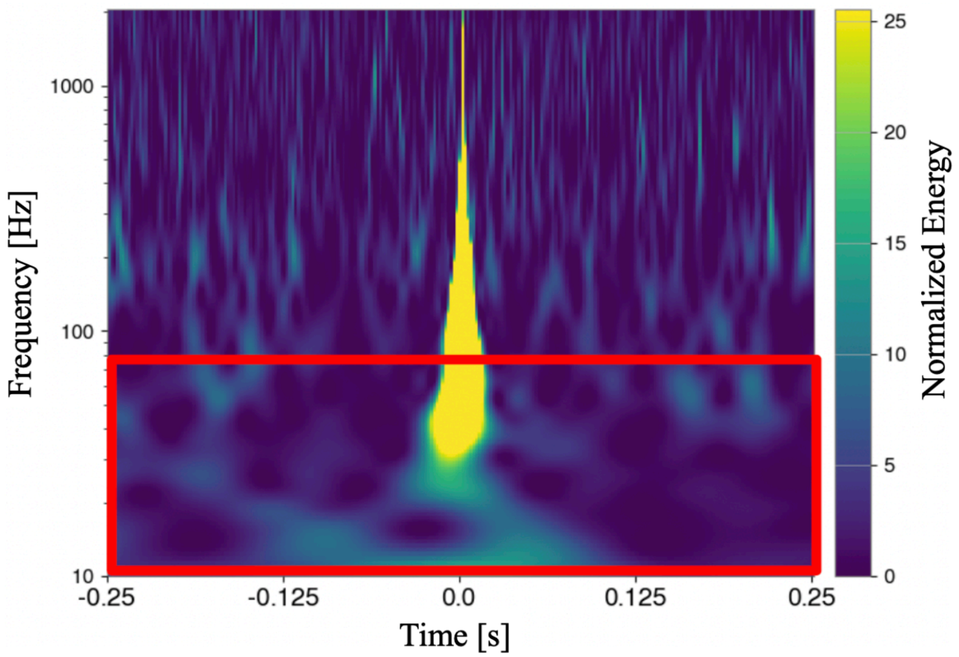


No subtraction applied

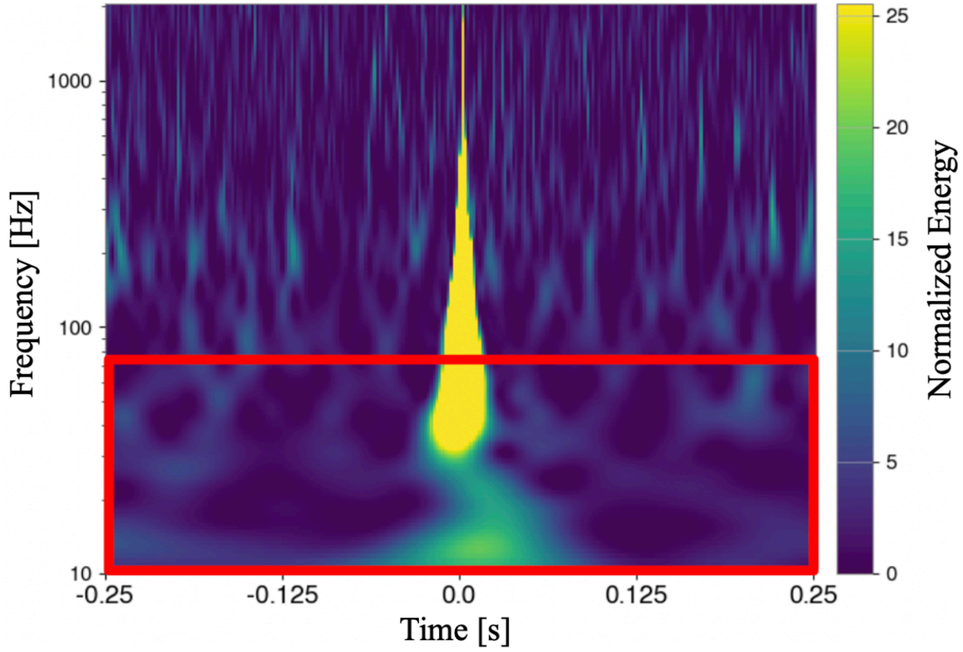


We expected O4 low-latency data to apply a non-linear subtraction of 60 Hz power artifacts.

Subtraction applied



No subtraction applied

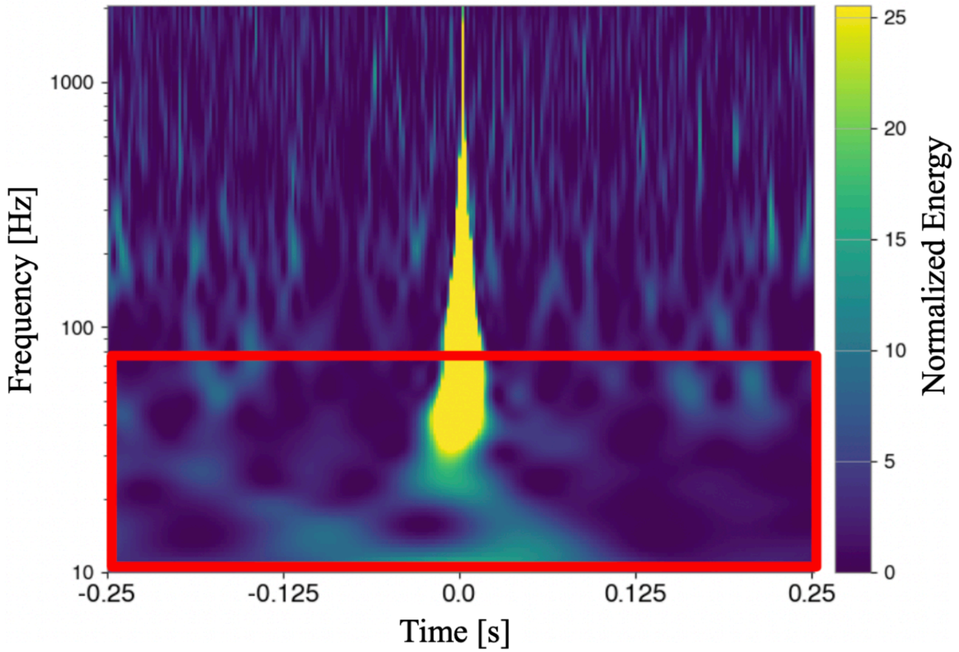


We did an experiment:

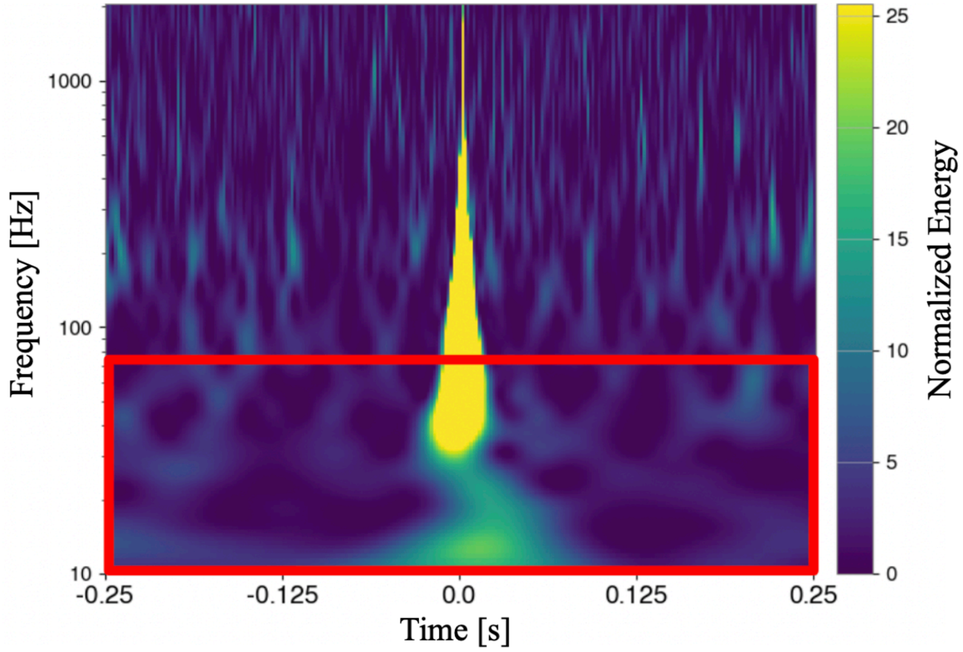
If GSpyNetTree is trained on right-like samples, how accurate are its predictions on left-like samples?

We expected O4 low-latency data to apply a non-linear subtraction of 60 Hz power artifacts.

Subtraction applied



No subtraction applied



We did an experiment:

If GSpyNetTree is trained on left-like samples, how accurate are its predictions on right-like samples?

!

When tested on these glitches, the pre-O4 version of GSpyNetTree reduced its glitch classification accuracy in ~20%

Part of the solution

- GSpyNetTree is really sensitive in changes in background noise.
- O4 GSpyNetTree includes data for both LIGO detectors using “normal” data and data applying the non-linear subtraction of 60 Hz AC power artifacts, to be robust to a broad array of noise.
- We also include Virgo glitches, but Virgo does not implement this subtraction.

?

Although we get great results in this version of GSpyNetTree… is this enough for the future?

What considerations should we take into account when building a **signal-vs-glitch classifier?**

1. Glitches come in various time-frequency morphologies. ✓

Three separate classifiers sorted via candidate mass

1.1 Some glitches are more common than others. ✓

Include most common glitches during O3/ER15.

1.2 How to account for short/long morphologies? ✓

Consider spectrograms of 4 different durations.

2. As detectors become more sensitive, glitches become more common. ✓

3. Effect of (changes in) background noise in spectrogram classification. ✓

Building a training set

A training set must be:

- Robust
- Balanced
- Hopefully large
- Should also apply data augmentation techniques

A training set must be:

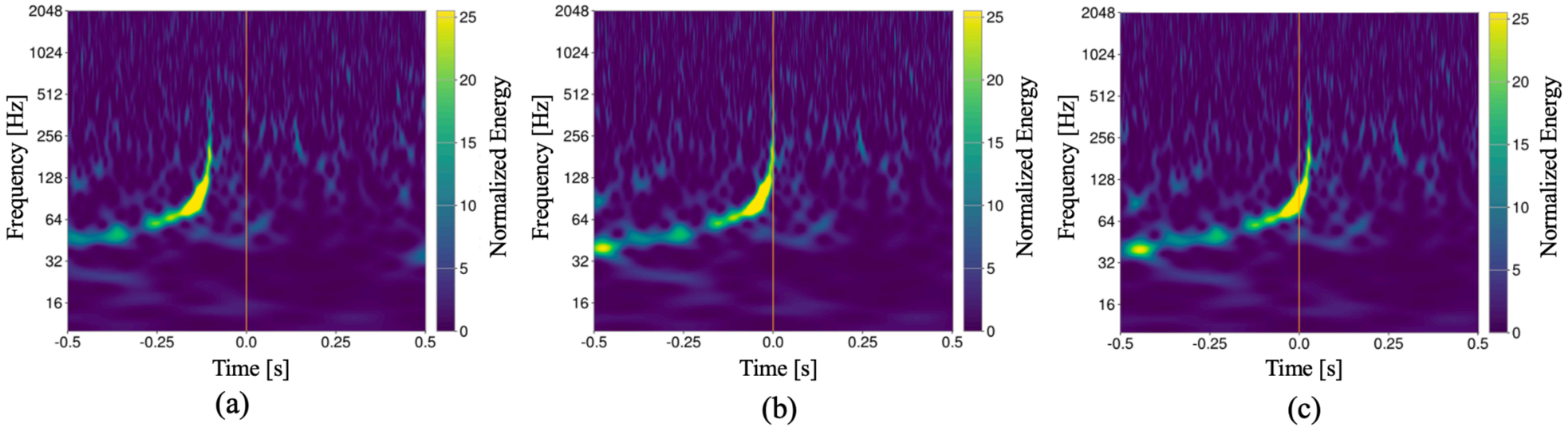
- Robust ✓
- Balanced* ✓
- Hopefully large ✓
- Should also apply data augmentation techniques ✓

*The Low-frequency Lines class is underrepresented, as we could only manually verify ~300 samples before O4 started.

Making GSpyNetTree robust to candidate

events not centered in $t = 0$ s.

- Including 4 extra time offsets (± 0.5 s).
- Data augmentation has proven useful for CNNs to generalize on a training set and avoid over-fitting.



Augmenting the training set

[14] Géron A. Hands on Machine Learning.

Results

Some key points

- For the purpose of GSpyNetTree, the most important thing is to **detect a glitch responsible for or in the presence of a GW.**

Some key points

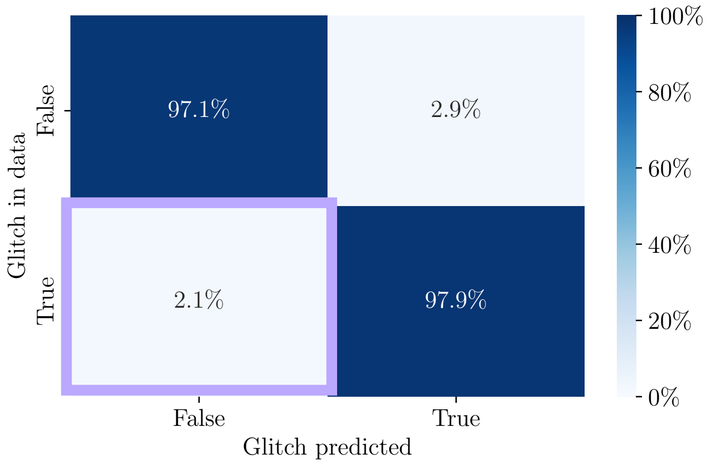
- For the purpose of GSpyNetTree, the most important thing is to **detect a glitch responsible for or in the presence of a GW.**
    - While not desirable, a glitch misclassified as another type of glitch is okay.
    - While not desirable, a GW misclassified as No Glitch is okay. Both cases mean no Data Quality issue was identified.

Some key points

- For the purpose of GSpyNetTree, the most important thing is to **detect a glitch responsible for or in the presence of a GW.**
    - While not desirable, a glitch misclassified as another type of glitch is okay.
    - While not desirable, a GW misclassified as No Glitch is okay. Both cases mean no Data Quality issue was identified.
- Worst case is:
    - **Glitch misclassified as GW/No Glitch (Data Quality issue missed).**
    - **False positive glitch (Glitch was not there but GSpyNetTree found one).**

Results on test data for glitches per classifier (using recall)

LM Classifier

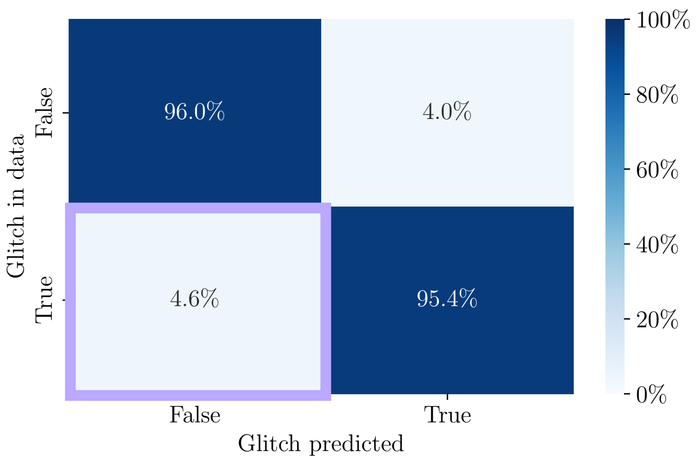


Only 2.1% of glitches are missed

HM Classifier

Only 2.3% of glitches are missed

EHM Classifier

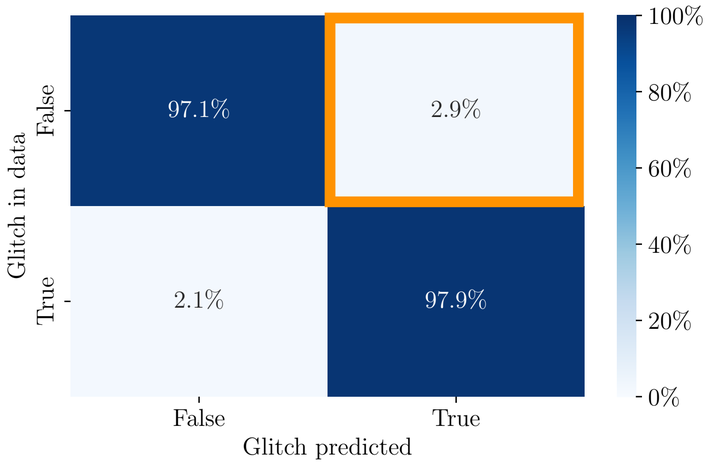


Only 4.6% of glitches are missed

In general, all three classifiers detect a high rate of glitches (>96%).

Results on test data for glitches per classifier (using recall)

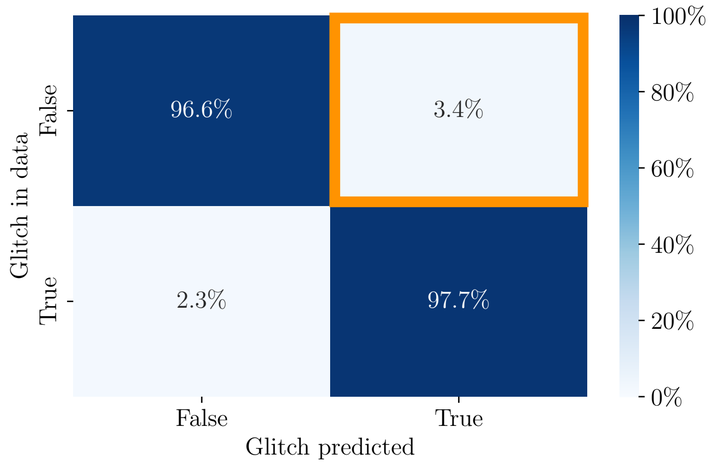
LM Classifier



Only 2.1% of glitches are missed

False alarm glitch rate: 2.9%

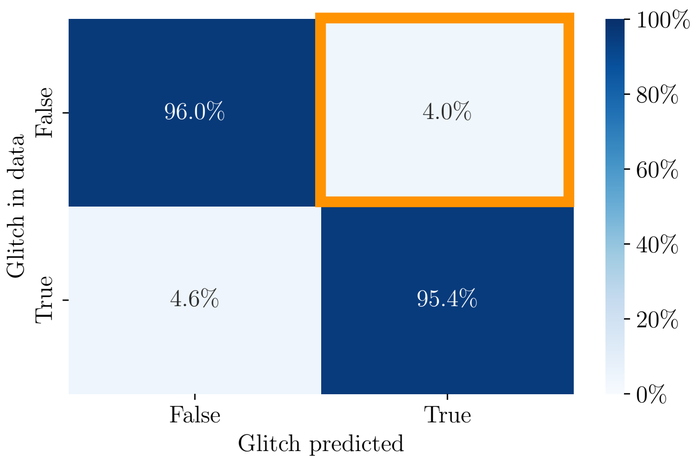
HM Classifier



Only 2.3% of glitches are missed

False alarm glitch rate: 3.4%

EHM Classifier



Only 4.6% of glitches are missed

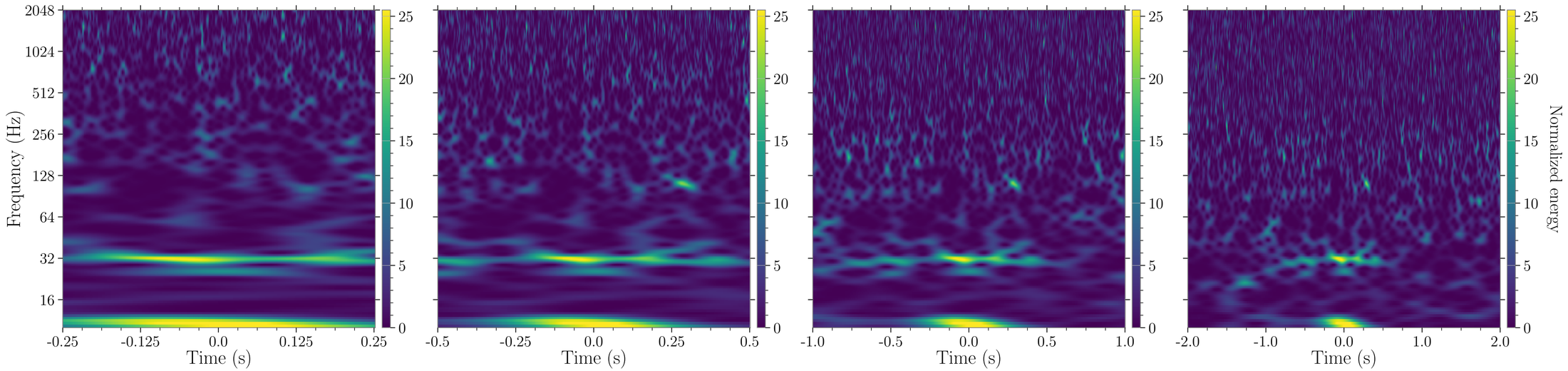
False alarm glitch rate: 4.0%

The false alarm is low, which makes GSpyNetTree's classifications reliable.

Other interesting results

- Due to its multilabel nature, GSpyNetTree can predict two or more glitches in an individual sample, even if it was not trained on them.

- The following sample was classified by the EHM classifier as Light Scattering ($f \approx 32$ Hz)+Low-frequency Lines ($f \approx 12$ Hz), while the original Gravity Spy classification was only Light Scattering.



Brief comment on O4a results

Some key points

- Results in under 2 minutes (after the analysis is triggered).

Some key points

- Results in under 2 minutes (after the analysis is triggered).
- In all significant events in which **BayesWave** was used to subtract a glitch, **GSpyNetTree's** correctly detected that glitch.

Some key points

- Results in under 2 minutes (after the analysis is triggered).
- In all significant events in which **BayesWave** was used to subtract a glitch, **GSpyNetTree's** correctly detected that glitch.
- All GSpyNetTree glitch detections were subsequently validated by 5-minute and 1-hour tier DQR tasks.

Some key points

- Results in under 2 minutes (after the analysis is triggered).
- In all significant events in which **BayesWave** was used to subtract a glitch, **GSpyNetTree's** correctly detected that glitch.
- All GSpyNetTree glitch detections were subsequently validated by 5-minute and 1-hour tier DQR tasks.
- GSpyNetTree correctly detected glitches in five of the six retracted events due to DQ issues in O4a.

More than one O4 candidate associated with glitches in close proximity of GWs

S230708t

S230708t is an event of interest because its false alarm rate, as estimated by the online analysis, is 4.3e-08 Hz, or about one in 8 months. The event's properties can be found at this URL:

https://gracedb.ligo.org/superevents/S230708t

The classification of the GW signal, in order of descending probability, is BBH (97%), Terrestrial (3%), NSBH (<1%), or BNS (<1%).

There was a high rate of noise transients (glitches) in the LIGO Hanford detector which may affect the parameters or the significance of the candidate.

More than one O4 candidate associated with glitches in close proximity of GWs

S230708t

S230708t is an event of interest because its false alarm rate, as estimated by the online analysis, is 4.3e-08 Hz, or about one in 8 months. The event's properties can be found at this URL:

https://gracedb.ligo.org/superevents/S230708t

The classification of the GW signal, in order of descending probability, is BBH (97%), Terrestrial (3%), NSBH (<1%), or BNS (<1%).

There was a high rate of noise transients (glitches) in the LIGO Hanford detector which may affect the parameters or the significance of the candidate.

These glitches detected by GSpyNetTree!

Shows full potential of multi-label architecture!

GSpyNetTree is only part of the solution

1. What happens when we test GSpyNetTree on Gravity Spy glitches not included in the original training set?

1. Testing GSpyNetTree on Gravity Spy glitches not included in the original training set

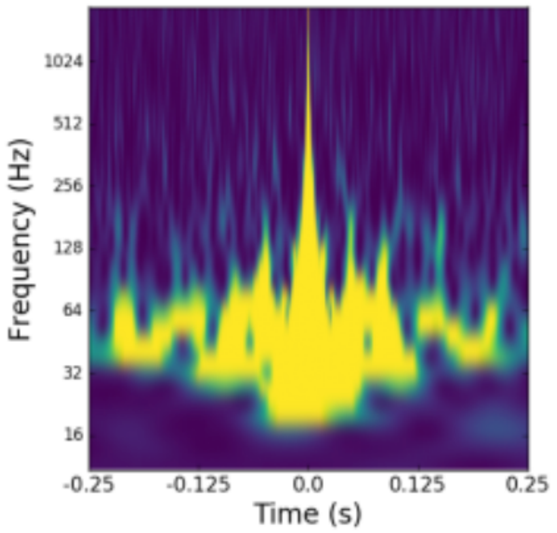
- Select 30 glitch examples from all Gravity Spy glitch classes not included in the original training set.
- We are only interested on whether a glitch is found or not.

1. Testing GSpyNetTree on Gravity Spy glitches not included in the original training set

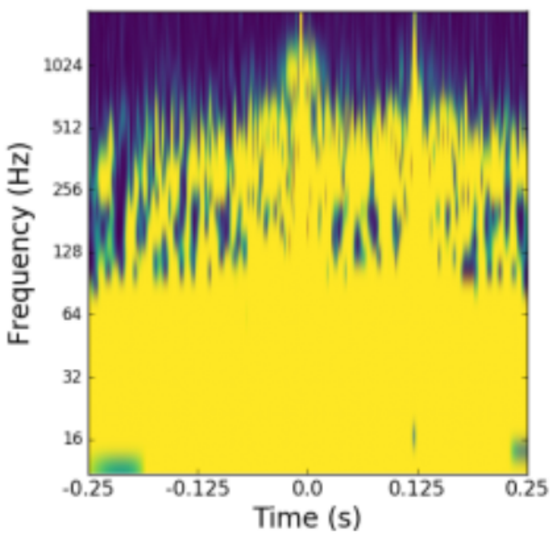
- Select 30 glitch examples from all Gravity Spy glitch classes not included in the original training set.
- We are only interested on whether a glitch is found or not.

Extremely loud glitches are morphologically similar to Koi Fish glitches (both are very loud): more than 98% accurate classifications.

Koi Fish



Extremely Loud



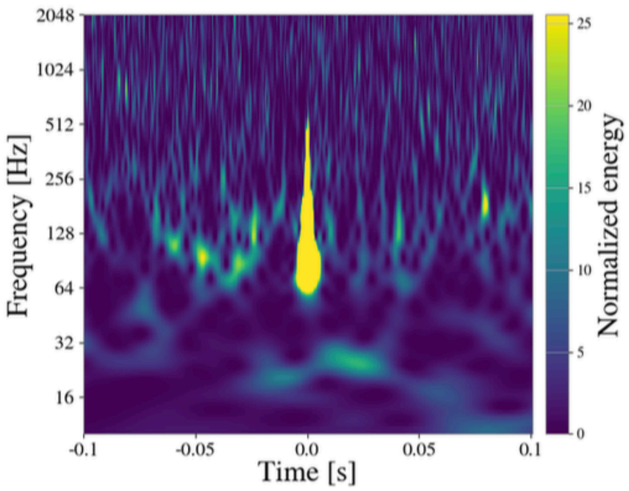
1. Testing GSpyNetTree on Gravity Spy glitches not included in the original training set

- Select 30 glitch examples from all Gravity Spy glitch classes not included in the original training set.
- We are only interested on whether a glitch is found or not.

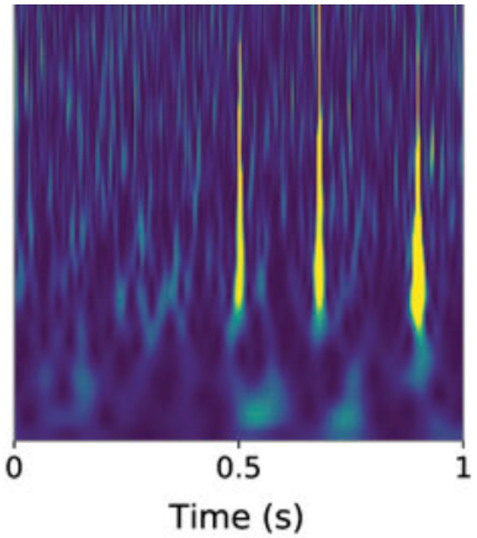
96% accurate glitch identification (but in a significant number of samples, one (or more) of the power artifacts mistaken as GW).

? How to identify multiple occurrences of the same glitch without flagging one of them as a GW?

Blip



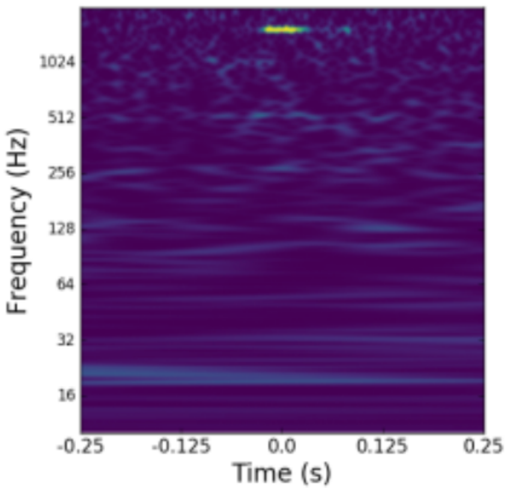
Repeating Blips



1. Testing GSpyNetTree on Gravity Spy glitches not included in the original training set

- Select 30 glitch examples from all Gravity Spy glitch classes not included in the original training set.
- We are only interested on whether a glitch is found or not.
  - Poor identification of high-frequency glitches: GSpyNetTree does not include any high frequency glitches.

1400 Ripples



? How to identify types of glitches not originally included in the training set without increasing the number of classes?

2. What happens when we test GSpyNetTree on O4 Virgo?

2. What happens when we test GSpyNetTree on Virgo?

- Consider a set of 200 O4 Virgo glitches.
- More than 90% correct glitch detections in three categories
- Less than 60% correct classifications for three other categories.

2. What happens when we test GSpyNetTree on Virgo?

- Consider a set of 200 O4 Virgo glitches.
- More than 90% correct glitch detections in three categories
- Less than 60% correct classifications (on average) for 3 categories.

GSpyNetTree is not transferable as-is to Virgo: while the architecture can be reused, the training sets must be tailored to each interferometer.

2. What happens when we test GSpyNetTree on Virgo?

- Consider a set of 200 O4 Virgo glitches.
- More than 90% correct glitch detections in three categories
- Less than 60% correct classifications (on average) for 3 categories.

But building the training sets requires a lot of person power.

GSpyNetTree is not transferable as-is to Virgo: while the architecture can be reused, the training sets must be tailored to each interferometer.

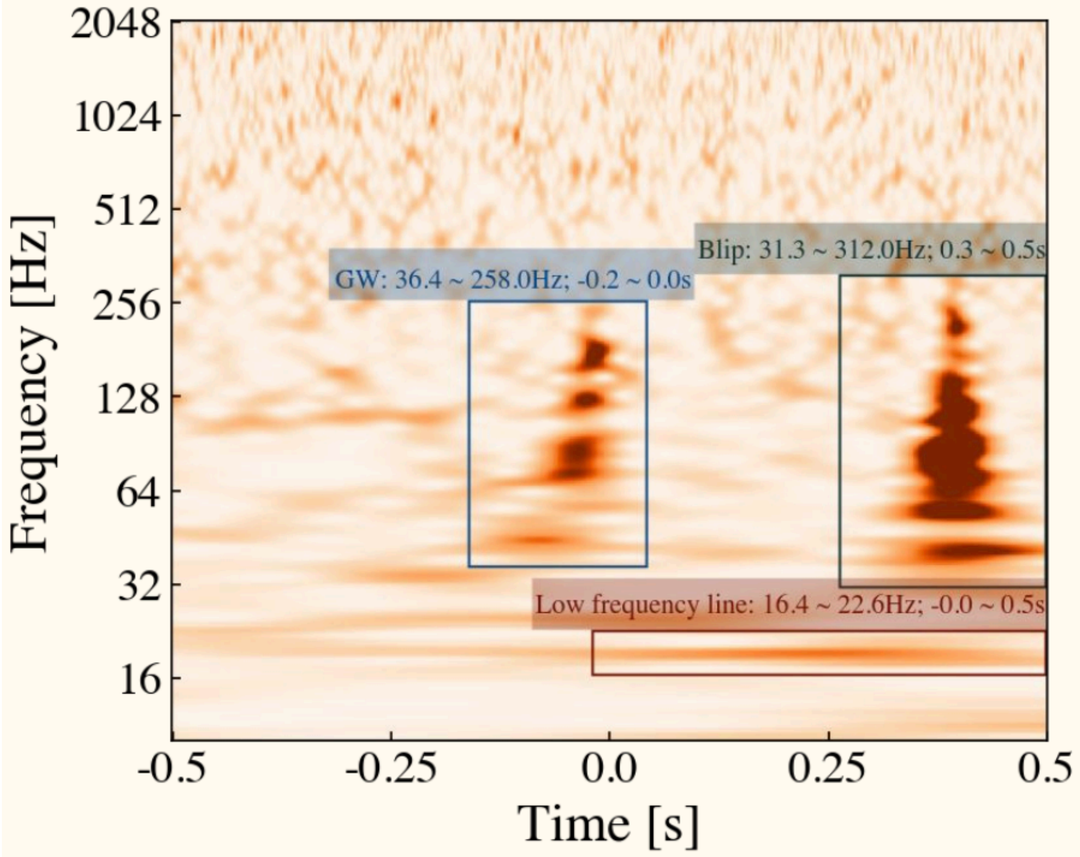
Future work: looking towards O5

How can we improve GSpyNetTree?

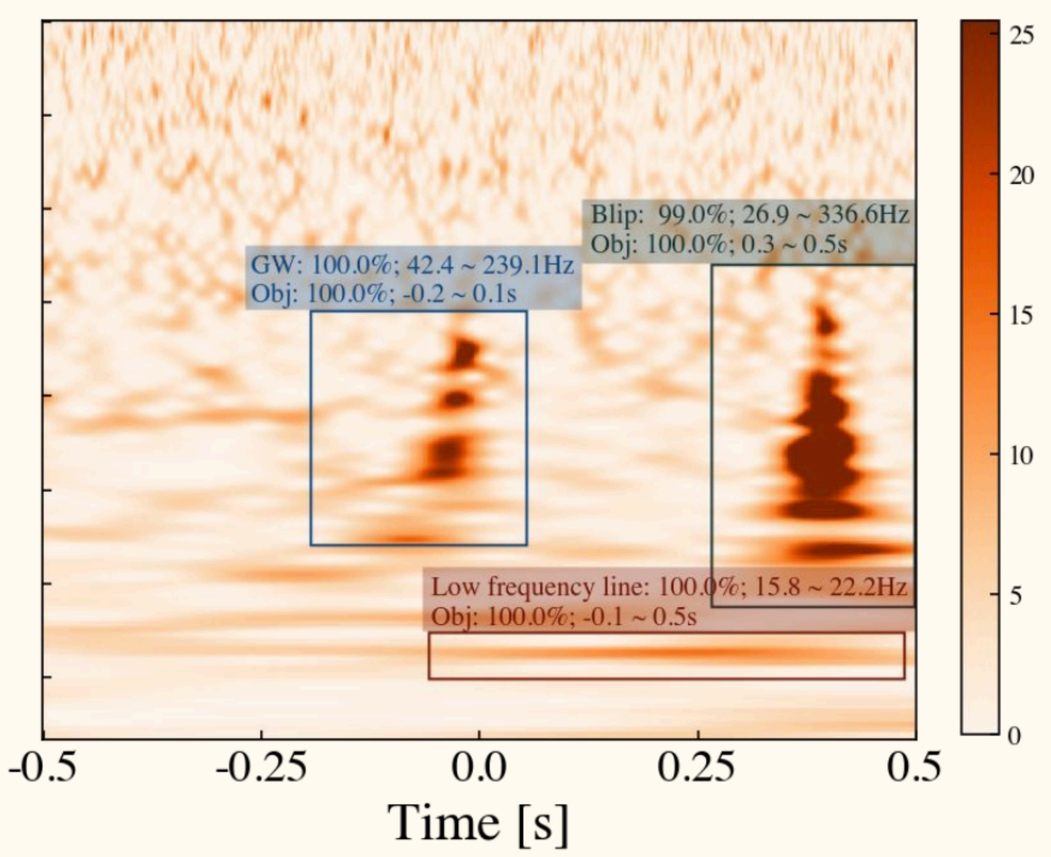
- GSpyNetTree is an ensemble of CNNs:
    - Uncertainty on output is hard to quantify.
    - Tend to be overconfident.
    - Possibility to extend to make Bayesian CNNs instead (currently led by Shreeja Bandyadhyay @ UBC).
- GSpyNetTree-S (Chan+25, in prep.)

Performance: IoU visualization

Ground truth - three regions of power excess



GSpyNetTreeS - three regions of power excess predicted



https://dcc.ligo.org/DocDB/0196/G2401842/001/GSpyNetTreeS.pdf

GSpyNetTree has been a very successful tool during O4!

Some open questions

?

Building training sets requires a lot of person power… is it time to evolve from supervised learning?

→

Background noise and glitches change over time.
Different detectors are under different conditions.
What if new glitches show up?

GSpyNetTree has been a very successful tool during O4!

Some open questions

?

Building training sets requires a lot of person power… is it time to evolve from supervised learning?

→

Background noise and glitches change over time.
Detectors are under different conditions.
What if new glitches show up?

Are spectrograms our best way to do signal-vs-glitch classification? Do we have anything better?

?

GSpyNetTree has been a very successful tool during O4!

Some open questions

?

Building training sets requires a lot of person power… is it time to evolve from supervised learning?

→

Background noise and glitches change over time.
Detectors are under different conditions.
What if new glitches show up?

?

How to automate the event validation process? (Especially in the future with higher event rate)?

Are spectrograms our best way to do signal-vs-glitch classification? Do we have anything better?

?

GSpyNetTree has been a very successful tool during O4!

Some questions to make our tool even more robust for future observing runs…

? Building training sets requires a lot of person power… is it time to evolve from supervised learning?

→ Background noise and glitches change over time. Detectors are under different conditions. What if new glitches show up?

? How to automate the event validation process? (Especially in the future with higher event rate)?

Are spectrograms our best way to do signal-vs-glitch classification? Do we have anything better? ?

We need your help to improve signal-vs-glitch classification!

? How to identify multiple occurrences of the same glitch without flagging one of them as a GW?

https://iopscience.iop.org/article/10.1088/1361-6382/ad2194

https://dcc.ligo.org/P2500368

This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459.