Generalized approach to matched filtering using neural networks

Jingkai Yan,[1,4] Mariam Avagyan,[1,4] Robert E. Colgan,[2,4] Doğa Veske,[3] Imre Bartos,[6] John Wright,[1,4] Zsuzsa Márka,[4,5] and Szabolcs Márka[3,4]

[1]*Department of Electrical Engineering, Columbia University in the City of New York, 500 W. 120th St., New York, New York 10027, USA*
[2]*Department of Computer Science, Columbia University in the City of New York, 500 W. 120th St., New York, New York 10027, USA*
[3]*Department of Physics, Columbia University in the City of New York, 538 W. 120th St., New York, New York 10027, USA*
[4]*Data Science Institute, Columbia University in the City of New York, 550 W. 120th St., New York, New York 10027, USA*
[5]*Columbia Astrophysics Laboratory, Columbia University in the City of New York, 538 W. 120th St., New York, New York 10027, USA*
[6]*Department of Physics, University of Florida, PO Box 118440, Gainesville, Florida 32611-8440, USA*

(Received 29 October 2021; accepted 13 January 2022; published 10 February 2022)

Gravitational wave science is a pioneering field with rapidly evolving data analysis methodology currently assimilating and inventing deep learning techniques. The bulk of the sophisticated flagship searches of the field rely on the time-tested matched filtering principle within their core. In this paper, we make a key observation on the relationship between the emerging deep learning and the traditional techniques: *matched filtering is formally equivalent to a particular neural network*. This means that a neural network can be constructed analytically to exactly implement matched filtering and can be further trained on data or boosted with additional complexity for improved performance. Moreover, we show that the proposed neural network architecture can outperform matched filtering, both with or without knowledge of a prior on the parameter distribution. When a prior is given, the proposed neural network can approach the statistically optimal performance. We also propose and investigate two different neural network architectures `MNet-Shallow` and `MNet-Deep`, both of which implement matched filtering at initialization and can be trained on data. `MNet-Shallow` has a simpler structure, while `MNet-Deep` is more flexible and can deal with a wider range of distributions. Our theoretical findings are corroborated by experiments using real LIGO data and synthetic injections, where our proposed methods significantly outperform matched filtering at false positive rates above $5 \times 10^{-3}\%$. The fundamental equivalence between matched filtering and neural networks allows us to define a "complexity standard candle" to characterize the relative complexity of the different approaches to gravitational wave signal searches in a common framework. Additionally, it also provides a glimpse of an intriguing symmetry that could provide clues on interpretability, namely how neural networks approach the problem of finding signals in overwhelming noise. Finally, our results suggest new perspectives on the role of deep learning in gravitational wave detection.

DOI: 10.1103/PhysRevD.105.043006

I. INTRODUCTION

The discovery of cosmic gravitational waves [1], the windfall of binary black hole (BBH) merger detections [2,3], and the spectacular insights that multimessenger astrophysics provided [4,5] revolutionized how we understand the Universe. This leap was due to multiple factors, from instrumental advances to computing breakthroughs. Emerging interferometric gravitational wave detectors, KAGRA [6], GEO600 [7], Virgo [8], and LIGO [9,10], played a critical role as they provided the technology [11–13] enabling signals to be extracted from ripples in Einstein's space-time [14,15]. Of course, as it is not sufficient to have data with faint cosmic signals buried in the noise, the community had to rely on exquisitely sensitive data analysis algorithms to extract transient signals from the noisy data. The bulk of the discoveries were made by two classes of powerful data analysis approaches, *excess power* [16–18] and *matched filtering* [19–25]. The flagship matched filtering methods [26–38] reached unprecedented sophistication and became the workhorse of the field [2,3]. Insightful work also exist on the extent of optimality, role of intrinsic parameters, and effect of non-Gaussian backgrounds [39–41]. There is more

than historical evidence on their algorithmic power [42], and they are also considered optimal [22] when searching for chirps of known shape [20,43–45] embedded in well-behaved Gaussian noise. Within the optimality and success lie limitations, as the data are significantly more complex [46,47] than Gaussian noise and many cosmic signals are not as well known as the BBH models that are being used in searches [48]. Therefore, it is critical that we both seek data analysis methods beyond the horizon of current techniques and rigorously understand the place of current techniques in the broader field of possible methods.

An abundance of prior works has been using deep learning methods for gravitational wave detection. Convolutional neural networks have been shown to be capable of identifying gravitational waves and their parameters from binary black holes and binary neutron stars, with a performance approaching the matched filtering search currently used by LIGO, Virgo and KARGA [49–73]. In addition, these machine learning (ML) methods can also be applied to glitches and noise transients identification [53,74–79], signal classification and parameter estimation [80–84], data denoising [85,86], etc. While these works exhibit neural networks that could approach the performance of matched filtering, they are still often applied as or considered "black box" models. This makes it challenging to evaluate the statistical evidence provided by neural networks and to incorporate that evidence in downstream analyses [87].

This paper is motivated by a critical observation, which we substantiate below: *matched filtering with a collection of templates is formally equivalent to a particular neural network*, whose architecture and parameters are dictated by the templates. This observation has precedents in the machine learning literature, where deep neural networks are sometimes viewed as hierarchical template matching methods, with signal-dependent, class-specific templates [88–94]. Here, we delineate a simple and explicit equivalence between matched filtering and particular neural networks, which can be constructed analytically from a set of templates. This equivalence lies in the algorithmic level and does not depend on specific problem formulations.

In order to study the potential performance gains of using neural networks, we formulate the gravitational wave detection problem abstractly as the detection of a parametric family of signals. Under this framework, we show that the analytically constructed networks can also be used as a principled starting point for learning from data, yielding signal classifiers with better performance than their initialization, namely "standing on the shoulder of giants." Such learning can be applied to scenarios both with or without a prior distribution on the parameters. In particular, when a prior distribution is given, we show that the learned neural network can (empirically) approach the statistically optimal performance.

We propose and investigate two different neural network architectures for implementing matched filtering, respectively, `MNet-Shallow` and `MNet-Deep`. The former has a simpler structure, while the latter is more flexible and can deal with a wider range of distributions. These learned classifiers have a number of additional advantages: they do not require prior knowledge of the noise distribution, can be adapted to cope with time-varying noise distributions, and suggest new approaches to computationally efficient signal detection. We conducted experiments using real LIGO data [95] in order to demonstrate the feasibility and power of neural networks in comparison to matched filtering, where we validate our findings empirically that neural networks via training can reach better performance. Finally, interpreting matched filtering and neural networks in a common framework also allows a clear comparison of their computational/storage complexities and statistical strengths, consequently making deep-learning less of a mystery.

The rest of the paper is organized as follows. Section II introduces the problem of parametric signal detection as an abstraction of the gravitational-wave detection problem and discusses the two formulations of the objective. Section III discusses matched filtering as an approach to solving the parametric detection problem, as well as its limitations. Section IV illustrates how neural network models can be applied in this problem, in a way that exactly implements matched filtering at initialization. Section V discusses the training process of neural network models, and in particular, how it is aligned with the parametric signal detection problem. In Section VI, we present experimental results on real LIGO data and synthetic injections. We discuss some further implications of this work in Sec. VII and conclude in Sec. VIII.

II. PARAMETRIC SIGNAL DETECTION

The problem of identifying gravitational waves [96] in a single gravitational-wave detector data stream [97] can be formulated as follows: we observe detector strain data $\boldsymbol{x} \in \mathbb{R}^n$ and wish to determine whether $\boldsymbol{x}$ consists of astrophysical signal plus noise, or noise alone. We can model possible astrophysical signals as belonging to a parametric family,

$$S_\Gamma = \{\boldsymbol{s}_{\boldsymbol{\gamma}} | \boldsymbol{\gamma} \in \Gamma\}, \tag{1}$$

where the parameters $\boldsymbol{\gamma}$ can represent properties of the objects that generate the gravitational wave, such as masses, orbits, and spins. We assume the signals are normalized to have unit power, namely $\|\boldsymbol{s}_{\boldsymbol{\gamma}}\|^2 = 1$ for all $\boldsymbol{\gamma}$. We model noise as a random vector $\boldsymbol{z} \in \mathbb{R}^n$, which is assumed to follow distribution ρ_0 and be probabilistically independent of the signal. In this notation, our goal becomes one of solving a hypothesis testing problem,

$$H_0 : \boldsymbol{x} = \boldsymbol{z}, \tag{2}$$

$$\text{or} \quad H_1 : \boldsymbol{x} = \boldsymbol{s}_\gamma + \boldsymbol{z} \text{ for some } \boldsymbol{\gamma} \in \Gamma. \tag{3}$$

Note that except for special cases, such as when the hypothesis H_1 is simple, or when the parameters associated with H_1 satisfy certain monotone conditions, we usually do not have a uniformly most powerful test [99].

Our broad goal is to identify decision rules $\delta : \mathbb{R}^n \to \{0, 1\}$ that (i) have good statistical performance and (ii) can be implemented efficiently. Our approach will start with analytically defined neural networks, which precisely replicate matched filtering, and then train these networks to optimize their statistical performance. We will give training approaches that are compatible with two classical frameworks for formalizing the performance decision rules δ: the *Neyman-Pearson* framework, in which the parameter $\boldsymbol{\gamma}$ is a random vector with known distribution ν, and the *minimax* framework, in which we control the worst performance over all possible choices of the parameter $\boldsymbol{\gamma}$.

A. Neyman-Pearson framework

In this setting, one assumes that $\boldsymbol{\gamma}$ is a random vector with probability distribution ν. With this distribution ν, we can then view H_1 as a simple hypothesis. The false positive rate (FPR) associated with the rule δ is

$$\text{FPR} = \mathbb{P}_z[\delta(\boldsymbol{z}) = 1]. \tag{4}$$

The false negative rate (FNR) at signal $\boldsymbol{s}_\gamma$ is

$$\text{FNR}_\gamma = \mathbb{P}_z[\delta(\boldsymbol{s}_\gamma + \boldsymbol{z}) = 0]. \tag{5}$$

The *overall* false negative rate is

$$\text{FNR} = \int \text{FNR}_\gamma \mathrm{d}\nu(\boldsymbol{\gamma}). \tag{6}$$

The Neyman-Pearson criterion seeks the optimal tradeoff between FNR and FPR,

$$\min_\delta \text{FNR subject to FPR} \leq \alpha, \tag{7}$$

where α is a user-specified significance level.

There is a classical closed form expression for the optimal test under the Neyman-Pearson criterion: if ρ_0 and ρ_1 are the probability densities of the signal $\boldsymbol{x}$ under hypotheses H_0 and H_1, respectively, then the optimal test is given by comparing the *likelihood ratio,*

$$\lambda(\boldsymbol{x}) = \frac{\rho_1(\boldsymbol{x})}{\rho_0(\boldsymbol{x})}, \tag{8}$$

to a threshold τ, which depends on the significance level α. An illustration of an example problem is shown in Fig. 1.

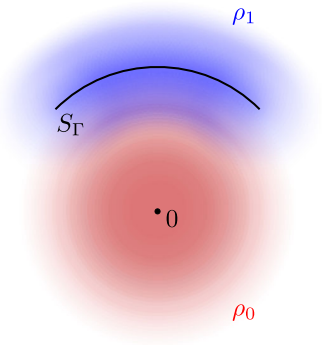

FIG. 1. An example of the parametric signal detection problem with signal space S_Γ. Densities ρ_0 and ρ_1 are shown in red and blue, respectively.

B. Minimax framework

When a good prior ν is not available or cannot be assumed, we can instead seek a decision rule that solves

$$\min \text{ WFNR subject to FPR} \leq \alpha. \tag{9}$$

at a given false positive rate, where WFNR is the *worst false negative rate* defined as

$$\text{WFNR} = \max_{\gamma \in \Gamma} \text{FNR}_\gamma. \tag{10}$$

In contrast to the Neyman-Pearson criterion, there is in general no simple expression for the minimax optimal rule δ [100]. In the next section, we will review matched filtering, a simple, popular approach to detection which is compatible with the minimax framework [albeit sub-optimal in terms of (9)], in the sense that it does not require a prior on $\boldsymbol{\gamma}$.

III. MATCHED FILTERING FOR PARAMETRIC DETECTION

Matched filtering is a powerful classical approach to signal detection, which applies a linear filter which is chosen to maximize the signal-to-noise ratio (SNR).

A. Optimality for single signal detection

In the simplest possible setting, in which (i) there is only one target signal $\boldsymbol{s}$, (ii) the observation $\boldsymbol{x}$ has the same length as $\boldsymbol{s}$, and (iii) the noise is uncorrelated (i.e., $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^*] = \sigma^2 \boldsymbol{I}$), matched filtering simply computes the inner product between the target $\boldsymbol{s}$ and the observation,

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \langle \boldsymbol{s}, \boldsymbol{x} \rangle \geq \tau. \tag{11}$$

When detecting a single signal $\boldsymbol{s}$ in iid Gaussian noise, this decision rule is optimal in both the Neyman-Pearson and

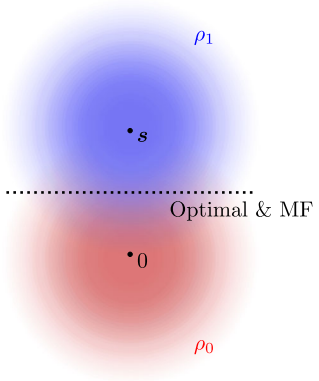

FIG. 2. Optimality of matched filtering in single signal detection.

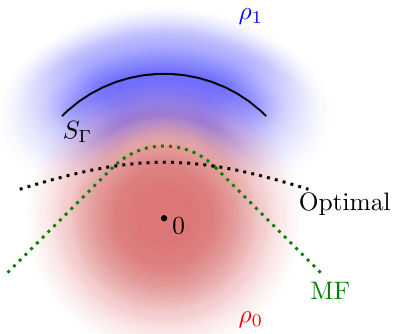

FIG. 3. Suboptimality of matched filtering under the Neyman-Pearson framework.

minimax senses: for example, if $z \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$, the likelihood ratio,

$$\lambda(\boldsymbol{x}) = \frac{\rho_0(\boldsymbol{x}-\boldsymbol{s})}{\rho_0(\boldsymbol{x})} = \exp\left(\frac{\langle \boldsymbol{s}, \boldsymbol{x}\rangle - \|\boldsymbol{s}\|^2/2}{\sigma^2}\right) \tag{12}$$

is a monotone function of $\langle \boldsymbol{s}, \boldsymbol{x}\rangle$, and so matched filtering implements the (optimal) likelihood ratio test. Figure 2 illustrates this optimality geometrically.

The simplicity and optimality in this setting make matched filtering a principled choice for signal detection and have inspired its application in settings that go far beyond the scope of this rigorous guarantee. In particular, the simplest and most practical extension of this rule to detecting parametric families of signals $\boldsymbol{s}_\gamma$ is suboptimal in both the Neyman-Pearson and minimax settings. Moreover, there are a number of additional factors that contribute to its suboptimality. These include unknown, non-Gaussian and possibly time-varying noise distributions as well as density and coverage issues in the template bank, which for complexity reasons may cover only a small portion of the phase space [22]. Nevertheless, we will see how matched filtering can inspire principled approaches to deriving more flexible decision rules which can address many of these challenges.

B. Extensions to parametric detection

The simplest extension of the decision rule (11) to *parametric* detection problems, in which there are multiple potential targets $\boldsymbol{s}_\gamma$, involves taking the maximum over the parameter space,

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \max_{\gamma\in\Gamma} \langle \boldsymbol{s}_\gamma, \boldsymbol{x}\rangle \geq \tau. \tag{13}$$

Here, we used the assumption that all templates have unit norm, namely $\|\boldsymbol{s}_\gamma\|_2^2 = 1,\ \forall\, \gamma \in \Gamma$. When this rule (13) is hard to implement in exact form, it can typically be approximated by taking samples $\boldsymbol{s}_{\gamma_1}, \ldots, \boldsymbol{s}_{\gamma_k}$ and setting

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \max_{i=1,\ldots,k} \langle \boldsymbol{s}_{\gamma_i}, \boldsymbol{x}\rangle \geq \tau. \tag{14}$$

When the sampling is sufficiently dense, the sampled matched filter rule (14) accurately approximates the ideal matched filter rule (13) [22]. This rule, while simple, is an important component of many sophisticated data analysis pipelines, including LIGO, Virgo and KARGA's template based searches for compact binary coalescence signals.

Note that the matched filtering decision rule (13) has connections to the (generalized) likelihood ratio test, where H_1 is the composite hypothesis $\boldsymbol{s}_\gamma \in S_\Gamma$. While this test has nice statistical properties, it is not guaranteed to be the uniformly most powerful test when the hypotheses are composite. For the rest of this paper, the term "likelihood ratio test" will be reserved for the test with a given prior and simple hypotheses, which satisfies the Neyman-Pearson criterion.

In contrast to the single signal setting, the simple extensions (13)–(14) of matched filtering to detecting parametric families of signals are not optimal: in the Neyman-Pearson setting, they do not achieve the minimal FNR for a given FPR, while in the minimax setting, they do not achieve the minimal WFNR for a given FPR.

The suboptimality of (13)–(14) under Neyman-Pearson can be observed by noting that the decision statistic $\max_\gamma \langle \boldsymbol{s}_\gamma, \boldsymbol{x}\rangle$ is not a monotone function of the likelihood ratio, which in iid Gaussian noise, for example, takes the form,

$$\lambda(\boldsymbol{x}) = \int \exp\left(\frac{\langle \boldsymbol{s}_\gamma, \boldsymbol{x}\rangle - \|\boldsymbol{s}_\gamma\|^2/2}{\sigma^2}\right) \mathrm{d}\nu(\boldsymbol{\gamma}). \tag{15}$$

Figures 3 and 4 illustrate such suboptimality for a particular problem configuration in $\mathbb{R}^2$. Note that throughout our paper, we will slightly abuse the term of receiver operating characteristic (ROC) curves by plotting FNR against FPR, instead of the convention of plotting FPR against true positive rate (TPR) $\equiv 1 -$ FNR. This highlights the connection to the notion of error rates in machine

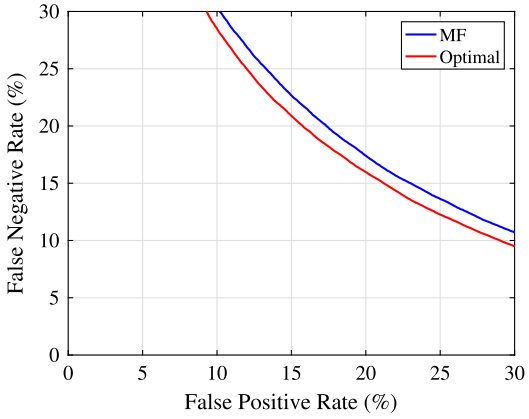

FIG. 4. Comparison of ROC curves of the optimal classifier and matched filtering in the two-dimensional concept as illustrated in Fig. 3.

learning and more importantly, facilitates demonstration of the curves and axis ranges at very low error rates.

It is, in a sense, unsurprising that matched filtering is suboptimal in this setting, since the decision rules (13)–(14) do not make use of the prior ν, while the likelihood ratio test assumes (and uses) this prior.

However, the matched filtering rule (13)–(14) is also in general suboptimal in the "prior-free" minimax setting. Consider the scenario in Fig. 5 as an example, where the signal space $S_\Gamma \subset \mathbb{R}^2$ consists of only two signals $\boldsymbol{s}_1 = [1,0]^T$ and $\boldsymbol{s}_2 = [0,1]^T$. Comparing the prior-free matched filtering decision rule δ_{MF} with the optimal decision rule δ_* under the Neyman-Pearson framework with uniform prior over the two signals, we see that δ_{MF} is suboptimal under Neyman-Pearson criterion with uniform prior. Moreover, from symmetry, it follows that for symmetric decision rules such as δ_{MF} and δ_* the worst FNR and the overall FNR are equal. This implies that δ_{MF} is also worse than δ_* under the minimax criterion.

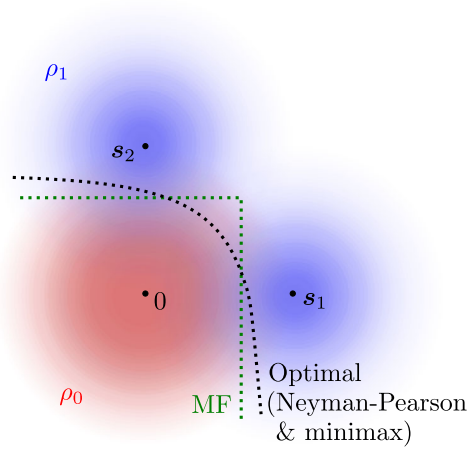

FIG. 5. Suboptimality of matched filtering under the minimax framework.

We also note that this suboptimality is, in some sense, not because we do not have sufficient templates. In the example shown in Fig. 5, the matched filtering model already covers the entire signal set which consists of two signals. Furthermore, we will see in the later discussions that matched filtering has other structural limitations when working with non-Gaussian noise distributions.

IV. FROM MATCHED FILTERING TO NEURAL NETWORKS

Since the matched filtering rule (14) is suboptimal for parametric detection, we will show that (i) the form of this rule suggests approaches to learning optimal rules for parametric detection, and (ii) the resulting classifiers have additional advantages, including greater flexibility and lower computational/storage complexity or cost. Our approach is driven by the observation: the matched filtering rule (14) is equivalent to a feed forward neural network.

A. Neural networks: Notation and basics

A *neural network* implements a mapping from the signal space $\mathbb{R}^n$ to an output space $\mathbb{R}^d$,

$$f_{\boldsymbol{\theta}}: \mathbb{R}^n \to \mathbb{R}^d. \tag{16}$$

Here, $\boldsymbol{\theta}$ represents the parameters of the network. Specifically, a fully connected neural network can be written as a composition of layers, each of which applies an affine mapping,

$$\boldsymbol{x} \mapsto \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \tag{17}$$

followed by an elementwise activation function ϕ,

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}^L \phi(\boldsymbol{W}^{L-1}\phi(\ldots\phi(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1)\ldots) + \boldsymbol{b}^{L-1}) + \boldsymbol{b}^L. \tag{18}$$

With slight abuse of notation, the activation function $\phi: \mathbb{R} \to \mathbb{R}$ acts elementwise when applied to a vector,

$$\phi([v_1, \ldots, v_n]^T) = [\phi(v_1), \ldots, \phi(v_n)]^T. \tag{19}$$

The intermediate products,

$$\boldsymbol{\alpha}^\ell(\boldsymbol{x}) = \phi(\boldsymbol{W}^\ell \phi(\ldots\phi(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1)\ldots + \boldsymbol{b}^\ell)), \tag{20}$$

are sometimes referred to as *features* [101]. In many situations, it is useful to "pool" features—this is especially useful for data with spatial or temporal structure; combining spatially adjacent features in a nonlinear fashion renders the decision more stable with respect to deformations of the input [102]. For example, *maximum pooling* takes the maximum of adjacent features. In our notation, we can denote this operation by ρ^ℓ and write

$$\boldsymbol{\alpha}^{\ell}(\boldsymbol{x}) = \rho^{\ell}\phi(\boldsymbol{W}^{\ell}\boldsymbol{\alpha}^{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}^{\ell}), \tag{21}$$

where the concise notation ρ^{ℓ} suppresses certain details about which features are combined. For clarity, we summarize this discussion in the following mathematical definition:

Definition 1: (Fully connected neural network) A fully connected neural network (FCNN) with *feature dimensions* $n^0,\ldots,n^L$, *preactivation dimensions* $m^1,\ldots,m^L$, *parameters*,

$$\begin{aligned}\boldsymbol{\theta} = (&\boldsymbol{W}^L \in \mathbb{R}^{m^L\times n^{L-1}},\ldots,\boldsymbol{W}^1 \in \mathbb{R}^{m^1\times n^0},\\ &\boldsymbol{b}^L \in \mathbb{R}^{m^L},\ldots,\boldsymbol{b}^1 \in \mathbb{R}^{m^1}),\end{aligned} \tag{22}$$

activation function $\phi:\mathbb{R}\to\mathbb{R}$ (extended to vector inputs by applying it elementwise), and *pooling operations* $\rho^{\ell}:\mathbb{R}^{m^{\ell}\to n^{\ell}}$ given by

$$[\rho^{\ell}]_i(\boldsymbol{v}) = \max_{j\in I_i^{\ell}} \boldsymbol{v}_j, \tag{23}$$

with $I_1^{\ell},\ldots,I_{n^{\ell}}^{\ell}$ being disjoint subsets of $[m^{\ell}]$, is a mapping $f_{\boldsymbol{\theta}}:\mathbb{R}^n\to\mathbb{R}^d$ defined inductively as $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \alpha^L(\boldsymbol{x})$ by setting $\boldsymbol{\alpha}^0(\boldsymbol{x}) = \boldsymbol{x}$, and

$$\boldsymbol{\alpha}^{\ell}(\boldsymbol{x}) = \rho^{\ell}\phi(\boldsymbol{W}^{\ell}\boldsymbol{\alpha}^{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}^{\ell}), \quad \ell = 1,\ldots,L. \tag{24}$$

When discussing neural networks, it is conventional to distinguish between the *network architecture*, which consists of the choices of feature dimensions n^{ℓ}, m^{ℓ}, activation function ϕ, and pooling operators ρ^{ℓ}, and the *network parameters* $\boldsymbol{\theta}$. Although we have stated a general definition, in specific architectures, the activation function ϕ and/or the pooling operators ρ^{ℓ} can be chosen to be trivial [$\phi(t) = t$ and/or $\rho^{\ell}(\boldsymbol{v}) = \boldsymbol{v}$].

Architectures. Neural networks are flexible function approximators [103]: universal approximation theorems indicate that *nonlinear* neural networks (with nonpolynomial activation ϕ) can accurately approximate any continuous function, as long as the network is sufficiently deep and/or wide [104–106]. There is a growing body of empirical and theoretical evidence showing that (relatively small) neural networks can learn relatively smooth functions over low-dimensional submanifolds of $\mathbb{R}^n$ with a complexity that is proportional to the manifold dimension, which in our problem equals the number of parameters in the parametrization $\boldsymbol{\gamma}\mapsto s_{\boldsymbol{\gamma}}$ [107].

Beyond these general considerations, there are scenarios in which the nature of the task dictates specific architectural choices. For example, in the field of inverse problems, neural network architectures can be generated by interpreting various optimization methods as taking on the structure in Definition 1 [108]. Our proposals will have a similar spirit, since they will interpret an existing method (matched filtering) as a particular instance of Definition 1.

Finally, a major architectural choice is whether to enforce additional structure on the matrices $\boldsymbol{W}^{\ell}$. When the input $\boldsymbol{x}$ is a time series, it is natural to structure the linear maps $\boldsymbol{\alpha}\mapsto\boldsymbol{W}\boldsymbol{\alpha}$ to be time invariant, i.e., to be convolution operators. To exhibit the equivalence between matched filtering and neural networks in the simplest possible setting, here we train our networks on injections whose starting time is fixed, and focus on fully connected neural networks (not enforcing convolutional structure).

In deployment, the input data is a time series, and astrophysical signals can occur at any time. In this setting, the matched filtering rule is applied in a sliding fashion. Similarly, the neural networks proposed here can be also deployed in a sliding fashion, which effectively converts them to particular convolutional networks. Both the equivalence between matched filtering and particular neural networks and the potential advantages of neural networks carry over to this setting.

Parameters. There are various approaches to choosing the network parameters $\boldsymbol{\theta}$. The dominant approach is to learn these parameters by optimization on data: one chooses initial parameters at random (with appropriate variance to ensure stability) and then iteratively adjusts them to best fit a given set of "training data." However, it is also possible in some scenarios to either (i) simply choose the weights at random, or (ii) to generate the weights analytically, either by connecting the network architecture to existing structures/algorithms [108] or from harmonic analysis considerations [109]. There are approaches that lie in between purely data-driven and purely analytical approaches to choosing $\boldsymbol{\theta}$. For example, it is possible generate initial weights analytically, and then tune them on training data. This hybrid approach achieves excellent performance on a number of inverse problems in imaging (super-resolution [110], magnetic resonance image reconstruction [111] etc.).

In the following sections, we will follow this approach: we will give two ways of interpreting the matched filtering decision rule (14) as a fully connected neural network, by making specific (analytical) choices of the architecture and parameters. These analytically chosen parameters can then be used as an initialization for learning on data. We will also see that in addition to this closed-form construction for equivalence, neural network models can be further trained on data to achieve improved performance.

B. Matched filtering as a shallow neural network

In the language of the previous section, it is not hard to express the decision statistic (14) of matched filtering as a specific fully connected neural network with one layer ($L = 1$). Writing

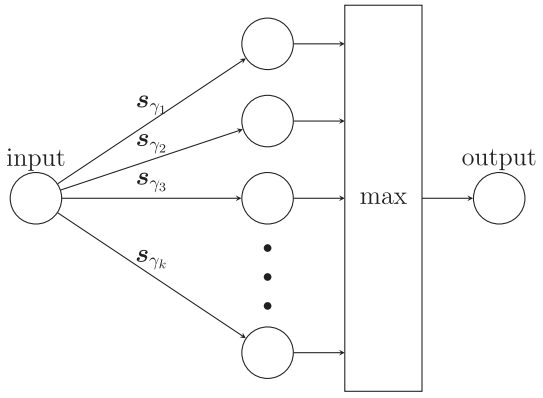

FIG. 6. Illustration of `MNet-Shallow`. Bias terms are omitted in the illustration. (We note that for more complex networks arbitrary pooling operations can replace the "max" box.)

$$\rho^1(\boldsymbol{z}) = \max_i z_i, \tag{25}$$

$$\phi(t) = t, \tag{26}$$

$$\boldsymbol{W}^1 = \begin{bmatrix} \boldsymbol{s}^*_{\gamma_1} \\ \boldsymbol{s}^*_{\gamma_2} \\ \vdots \\ \boldsymbol{s}^*_{\gamma_k} \end{bmatrix} \in \mathbb{R}^{k \times n}, \tag{27}$$

$$\boldsymbol{b}^1 = \boldsymbol{0}, \tag{28}$$

([112]), we have

$$\max_i \langle \boldsymbol{s}_{\gamma_i}, \boldsymbol{x} \rangle = \rho^1 \phi(\boldsymbol{W}^1 \boldsymbol{x} + \boldsymbol{b}^1). \tag{29}$$

In words, the features produced by this neural network correspond to the correlations of the input with the templates $\boldsymbol{s}_{\gamma_1}, \ldots, \boldsymbol{s}_{\gamma_k}$. Figure 6 illustrates this (simple) architecture, which we label `MNet-Shallow`.

Where needed below, we refer to the input-output relationship implemented by this architecture as

$$f_{\texttt{MNet-Shallow},\boldsymbol{\theta}}(\boldsymbol{x}), \tag{30}$$

where $\boldsymbol{\theta} = (\boldsymbol{W}^1, \boldsymbol{b}^1)$ represent the weights and biases. When these are chosen as in (27)–(28), `MNet-Shallow` implements the matched filtering decision rule. We note that these weights can be constructed analytically based on the given templates.

By learning the weights $\boldsymbol{W}^1$ and biases $\boldsymbol{b}^1$ from examples, we can further adapt this network to implement a more general family of decision rules, beyond matched filtering (14) with templates $\boldsymbol{s}_\gamma$. Nevertheless, there are limitations to this architecture. Notice that in `MNet-Shallow` there is only one layer of affine operations, and so this architecture does not satisfy the dictates of the universal approximation theorem [105,113].

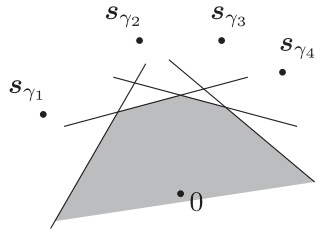

FIG. 7. The set of points classified as noise by matched filtering and `MNet-Shallow` is always a convex set.

More geometrically, we can notice that the decision rule associated with `MNet-Shallow` is a maximum of affine functions. This means that for any choice of $\boldsymbol{W}^0$ and $\boldsymbol{b}^0$, the decision boundary is the boundary of a convex set. This property is also true for matched filtering, which shares exactly the same form. An illustration of this property is shown in Fig. 7.

How restrictive is this limitation? In the context of parametric detection, this depends largely on the noise distribution. If the noise is Gaussian, the optimal decision boundary is itself the boundary of a convex set:

Proposition 2: Suppose that the noise $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Then for any significance level α, the optimal (Neyman-Pearson) decision region,

$$\{\boldsymbol{x} | \lambda(\boldsymbol{x}) \leq \tau\}, \tag{31}$$

is a convex subset of $\mathbb{R}^n$, where τ is a constant determined by the significance level α.

Proof.—Please see the Appendix. ∎

However, for general non-Gaussian distributions, the optimal decision region is often nonconvex. We illustrate this result in Fig. 8. In fact, this suggests an intrinsic structural limitation of matched filtering and similar architectures. Since in reality the noise distribution is not perfectly Gaussian, we cannot expect the optimal decision region to be convex, and hence, the matched filtering structure is unable to approach the performance of the likelihood ratio test with arbitrary precision, even if any number of templates (including ones outside the original signal space) are allowed. In such cases, we can benefit from using a more flexible architecture, which we now introduce.

C. Matched filtering as a deep neural network

We describe an alternative way of expressing template matching as a neural network, which leads to deep, nonlinear architectures that are more flexible than `MNet-Shallow`. We label this structure `MNet-Deep`. In this architecture, we do not compute the maximum in a straightforward way using pooling. Instead, we propose an alternative architecture which is more flexible, and can

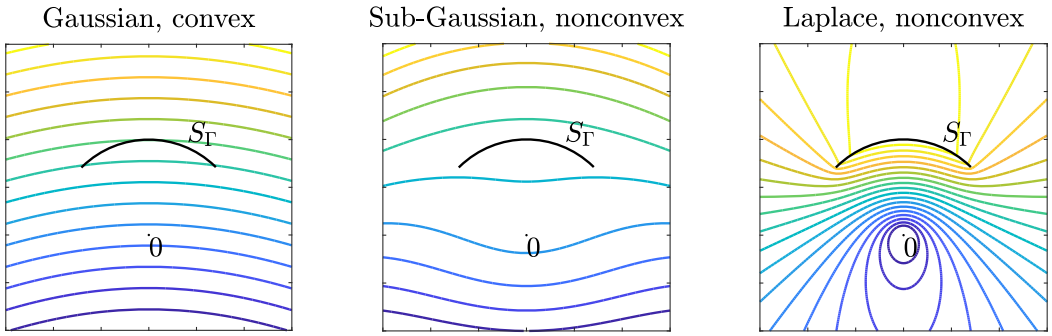

FIG. 8. Contours of log likelihood ratio with various noise distributions, and whether the optimal decision regions with $\delta = 0$ is always convex. Yellow represents larger values, and blue represents lower values. From left to right: (1) Gaussian distribution, convex; (2) Sub-Gaussian distribution $\rho_{\text{noise}}(\boldsymbol{x}) \propto \exp(-C\|\boldsymbol{x}\|^3)$, not necessarily convex; (3) Laplace distribution, not necessarily convex.

approximate a wider class of functions. In particular, we will no longer be restricted to implementing decision boundaries that are boundaries of convex sets, allowing us to handle scenarios with non-Gaussian noise. An illustration of this `MNet-Deep` is shown in Fig. 9.

Our construction is based on the rectified linear unit (ReLU) nonlinearity,

$$\phi(t) = \max(t, 0). \tag{32}$$

This is arguably the most commonly used nonlinearity function in modern deep learning.

The matched filtering decision rule takes the maximum of a family of linear functions $\langle \boldsymbol{s}_{\gamma_i}, \boldsymbol{x} \rangle$. Instead of simply "pooling" these functions as in the previous section, we implement the maximum operation using compositions of ReLUs and linear operations. In particular, observe that the maximum of two numbers can be written as a linear combination of three ReLU units,

$$\max(a,b) = b + \phi(a-b) = \phi(b) - \phi(-b) + \phi(a-b). \tag{33}$$

The basic idea is to create a hierarchical structure of such 3-ReLU-units; each of which takes a pairwise maximum of its inputs. Our `MNet-Deep` construction will perform convolutions with the templates $\boldsymbol{s}_{\gamma_i}$, followed by this hierarchical structure for computing the maximum.

Figure 10 illustrates this hierarchical structure for the particular example of four inputs. The network in Fig. 10 can be expressed as a ReLU network, with sparse weight matrices $\boldsymbol{W}^\ell (\ell = 0, 1, 2)$ for the layers, respectively,

$$\boldsymbol{W}^0 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & -1 \end{bmatrix}, \qquad \boldsymbol{W}^2 = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}, \tag{34}$$

$$\boldsymbol{W}^1 = \boldsymbol{W}^0 \otimes \boldsymbol{W}^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{35}$$

Generalizing this construction, we obtain a network that takes the maximum of k numbers, using $\lceil \log_2 k \rceil + 1$ layers.

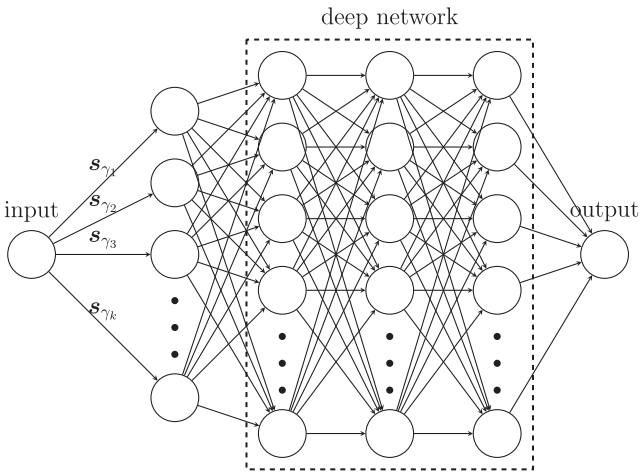

FIG. 9. Illustration of `MNet-Deep`. Bias terms are omitted in the illustration. This network structure is obtained by replacing the max module in matched filtering (as in Fig. 6) with a deep network.

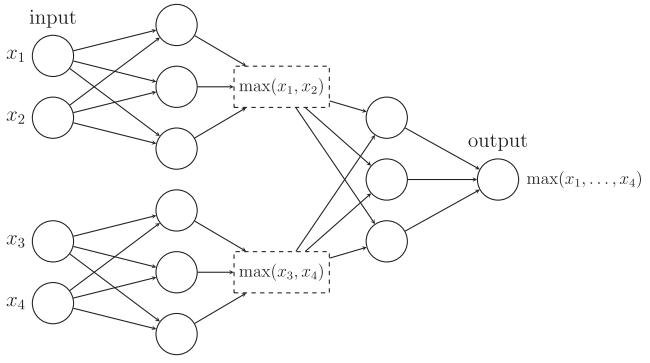

FIG. 10. Illustration of implementing max with a ReLU network. The dashed boxes in the middle are not actual nodes in the network, but "imaginary" nodes to facilitate construction.

While the example above delineates a precise form of the ReLU network, this approach can in fact be made flexible. To ensure that the network output is indeed the maximum of the k inputs, we must ensure that at each layer, each feature participates in at least one of the pairwise max operations. This means that at layer ℓ, we must have at least $k/2^{\ell}$ features. However, we are free to add more intermediate features, with additional (redundant) max operations. This does not change the output of the network, but it affords additional flexibility when we attempt to train the network on data. In particular, this allows the construction of arbitrarily wide or deep ReLU networks and can therefore approximate any regular continuous function [105,113].

There is also a degree of freedom in choosing which features participate in each pairwise maximum operation, which could be chosen in various ways. In our implementation, we use the following way to pair up the nodes in layer l for pairwise maximum operations that get to layer $l+1$. Assume layer l contains $2p$ nodes. First, pair up the nodes with consecutive indices, namely pair up node $2i-1$ with node $2i$ for $i=1,\ldots,p$. This ensures that each node is covered by at least one maximum operation. After that, for each leftover node in layer $l+1$, we establish the corresponding pair in layer l by choosing the nodes at random in layer l. In the following, we label this network `MNet-Deep`. We emphasize for clarity that the nodes between consecutive layers are fully connected in the neural network; however, the weights not associated with pairwise maximum operations are all initialized to zero. Below, where needed we refer to the decision rule associated with this network as

$$f_{\texttt{MNet-Deep},\boldsymbol{\theta}}(\boldsymbol{x}), \tag{36}$$

where $\boldsymbol{\theta}$ represent the collection of all weights and biases. The above discussion again gives a recipe for choosing these weights analytically such that the decision rule for `MNet-Deep` coincides with the matched filtering rule.

In contrast to `MNet-Shallow`, `MNet-Deep` is a more flexible architecture. In particular, this architecture satisfies the dictates of the universal approximation theorem. Geometrically, it is not restricted to convex decision regions, which makes it capable of achieving optimal decision boundaries even when the noise is heavy-tailed or has other nonideal properties.

D. Equivalence of matched filtering and neural networks

We have demonstrated by construction the following claim:

Given any collection of templates $\boldsymbol{s}_{\gamma_1},\ldots,\boldsymbol{s}_{\gamma_k}$ *(for any* $k\geq 1$*), one can analytically determine weights* $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_d$ *such that*

$$f_{\texttt{MNet-Shallow},\boldsymbol{\theta}_s}(\boldsymbol{x}) = \max_{i=1\ldots k}\langle \boldsymbol{s}_{\gamma_i},\boldsymbol{x}\rangle \tag{37}$$

$$f_{\texttt{MNet-Deep},\boldsymbol{\theta}_d}(\boldsymbol{x}) = \max_{i=1\ldots k}\langle \boldsymbol{s}_{\gamma_i},\boldsymbol{x}\rangle \tag{38}$$

for all $\boldsymbol{x}\in\mathbb{R}^n$.

We emphasize the complete generality of this claim: it holds for any number and choice of templates. Moreover, it does not depend on training: the networks can be constructed analytically to implement the matched filtering rule. Nevertheless, we will see in the next section that they can be further adapted based on observed data to strictly outperform matched filtering, in terms of the Neyman-Pearson criterion.

The equivalence between matched filtering and particular neural networks has an additional conceptual advantage: it allows for a clear comparison of the resource complexity of different search methods, in terms of storage and computation. This is valuable because different methods may cut out very different tradeoffs between complexity and accuracy/performance. Neural network implementations of matched filtering can be viewed as "complexity standard candles" against which the performance of more sophisticated networks can be measured. In particular, the complexity of a neural network model may be quantified by the total number of nodes (neurons) in the network, which approximately characterizes the number of elementary operations performed for evaluating an input instance [114,115]. We will look for the most appropriate measure of complexity for this problem, and provide detailed analysis in future studies.

V. TRAINING TO APPROACH STATISTICAL OPTIMALITY

In the previous section, we gave two ways of analytically constructing neural networks that reproduce the matched filtering decision rule and hence, exhibit exactly the same performance as matched filtering. The major advantage of this interpretation of matched filtering is that the resulting model can be further trained on sample data to improve its statistical performance or adapt it to handle non-Gaussian noise distributions, or in other words "standing on the shoulder of giants." In a typical neural network training problem, we have access to labeled samples,

$$(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N), \tag{39}$$

each of which consists of an observation $\boldsymbol{x}_i\in\mathbb{R}^n$ and a corresponding label $y_i\in\{0,1\}$, which indicates whether $\boldsymbol{x}_i$ contains a noisy signal ($y_i=1$) or noise only ($y_i=0$). To date, we have only a moderate number of confirmed gravitational wave detections and hence, have far more negative examples than positive examples. We address this issue by generating our positive training examples by injecting synthetic waveforms into (real) LIGO noise

strains. Below, we describe two different training schemes, motivated by the Neyman-Pearson and minimax criteria, which leverage this data to perform training of the neural networks.

Training for Neyman-Pearson. In this setting, we assume that the prior ν is known, and generate positive examples by first sampling $\boldsymbol{\gamma}_i \sim \nu$, and setting $\boldsymbol{x}_i = \boldsymbol{s}_{\boldsymbol{\gamma}_i} + \boldsymbol{z}_i$, where $\boldsymbol{z}_i$ is observed LIGO noise strain. We solve the following optimization problem:

$$\min_{\boldsymbol{\theta}} \mathcal{R}_N(f_{\boldsymbol{\theta}}) \coloneqq \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i). \tag{40}$$

Here, the *loss function* $\ell(\hat{y}, y)$ measures the misfit between the predicted label $\hat{y}$ and the true label y. Typical choices include the square loss $(\hat{y} - y)^2$ and the logistic loss,

$$y \log(f_{\text{sigmoid}}(\hat{y})) + (1 - y)\log(1 - f_{\text{sigmoid}}(\hat{y})), \tag{41}$$

where $f_{\text{sigmoid}}(\cdot)$ denotes the logistic/sigmoid function,

$$f_{\text{sigmoid}}(x) = \frac{1}{1 + \exp(-x)}. \tag{42}$$

Is this training strategy compatible with the Neyman-Pearson criterion? The following proposition answers this question in the affirmative. Consider the following setup: training data $(\boldsymbol{x}_i, y_i)$ are generated independently at random, by setting $y_i = 1$ with probability $p \in (0, 1)$ and choosing $\boldsymbol{x}_i = \boldsymbol{s}_{\boldsymbol{\gamma}_i} + \boldsymbol{z}_i$ when $y_i = 1$ and $\boldsymbol{x}_i = \boldsymbol{z}_i$ when $\boldsymbol{y}_i = 0$, with $\boldsymbol{\gamma}_i \sim \nu$, and $\boldsymbol{z}_i \sim \rho_{\text{noise}}$. Let

$$\mathcal{R}_\infty(f) = \mathbb{E}_{(\boldsymbol{x},y)}\ell(f(\boldsymbol{x}), y). \tag{43}$$

This represents the large-sample limit of $\mathcal{R}_N$: as $N \to \infty$, $\mathcal{R}_N(f) \to \mathcal{R}_\infty(f)$. The following proposition shows that the population risk $\mathcal{R}_\infty$ is minimized by (a monotone function of) the likelihood ratio λ:

Proposition 3: Suppose that for any $y = 0, 1$, the loss $\ell(\hat{y}, y)$ is a strictly convex differentiable function of $\hat{y}$ that is minimized at $\hat{y} = y$. [116] Then the unique optimal solution $f_\star$ to the (functional) optimization problem,

$$\min_f \mathcal{R}_\infty(f), \tag{44}$$

is a strictly increasing function of the likelihood ratio λ,

$$f_\star(\boldsymbol{x}) = g(\lambda(\boldsymbol{x})), \tag{45}$$

where g is a strictly increasing function that depends on ℓ.

Proof.—Please see Appendix. ∎

This result can be interpreted as saying: *"a sufficiently flexible classifier, trained on a sufficiently large dataset will produce the optimal decision rule."* Hence, training to minimize the empirical risk $\mathcal{R}_N(f_{\boldsymbol{\theta}})$ is compatible with the Neyman-Pearson criterion.

While this is a promising observation, we should keep in mind a number of remaining issues: How much data are required? What are effective approaches to minimizing the empirical risk $\mathcal{R}_N$? In the next section, we investigate these questions experimentally.

Training for minimax. In this setting, we do not assume any prior, and aim to minimize the worst false negative rate using the formulation in (9). We convert the constrained problem (9) to an equivalent unconstrained problem,

$$\min_\delta \max_{\boldsymbol{\gamma}\in\Gamma} \text{FNR}_{\boldsymbol{\gamma}} + c \cdot \text{FPR}, \tag{46}$$

where c is a constant that depends on α. For tractability, we will fix c at a constant value to obtain a concrete optimization objective, and here, we fix $c = 1$. In actual deployment where a target significance level α is specified, we can also choose c at the level that corresponds to the specified α. Also, we sample the parameter space Γ at points $\{\boldsymbol{\gamma}_i\}_{i=1}^N$. Since FPR does not depend on $\boldsymbol{\gamma}$, it can be moved inside the maximization. Therefore, the minimax optimization problem can be transformed into

$$\min_\delta \max_{i=1,\ldots,N} (\text{FNR}_{\boldsymbol{\gamma}_i} + \text{FPR}). \tag{47}$$

This suggests a natural approach to training under the minimax criterion using first-order optimization methods. At each iteration, we estimate FPR and $\text{FNR}_{\boldsymbol{\gamma}_i}$ for each $i = 1, \ldots, N$, and choose i_* with the highest $\text{FNR}_{\boldsymbol{\gamma}_i}$. We then aim to reduce $\text{FNR}_{\boldsymbol{\gamma}_i} + \text{FPR}$, which can be estimated by using a sample dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ as

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \neq y_i], \tag{48}$$

where in the dataset, all $\boldsymbol{x}_i$ with corresponding $y_i = 1$ are generated specifically with signal parameter $\boldsymbol{\gamma}_{i_*}$, and half of data pairs in the dataset have $y_i = 0$. Finally, it is customary in optimization to replace the nondifferentiable 0-1 loss with a smooth loss function ℓ, and hence, we get the following risk minimization objective:

$$\frac{1}{N}\sum_{i=1}^{N} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i). \tag{49}$$

This expression is similar to (40), but the difference is that all positive data in the dataset here are associated with signal parameters $\boldsymbol{\gamma}_{i_*}$.

VI. SIMULATIONS AND EXPERIMENTS

A. Data generation

Data-driven methods such as neural networks typically require a large amount of data for training. The question of data sufficiency is especially acute in gravitational wave astronomy: we have only a moderate number of confirmed detections to date. We address this issue by generating our positive training examples by injecting synthetic waveforms into LIGO noise strains [95], which we elaborate below.

For LIGO noise data, we use the L1 strain from LIGO O2 run between August 1 and August 25, 2017, with ANALYSIS_READY segments only. The announced confident detections GW170809, GW170814, GW170817, GW170818, and GW170823 are removed from the strain, such that the data is at least 300 seconds away from these events. We used a total of 338 frame files each of 4096 seconds long, namely a total of 384.57 hours. The strain data are downsampled from the original 4096 Hz to 2048 Hz for processing efficiency. The downsampled L1 strain data are divided into segments of length 0.6 second, with each successive segment overlapping 50% of the previous segment.

We generate synthetic gravitational wave signals using PyCBC [26–32], with the following parameters. *Approximant:* IMRPhenomD. *Mass range:* 40 to 50 $M_\odot$, uniformly distributed. *Spin:* 0. *Sampling rate:* 2048 Hz. *Low frequency cutoff:* 30 Hz. *Coalescence phase:* 0. *Polarization:* plus [117]. With this specified mass range, at least 99.5% of the energy of the signal lies in an interval of length 0.3 second after preprocessing. We note that although the templates are not chosen uniformly in actual LIGO deployment [42,43,118–120], we make this choice here due to simplicity, and also the fact that the large number of templates make up for the possibly suboptimal choice of templates.

The above data are used to generate training and test datasets of positive and negative labeled data as follows. We divide the collection of downsampled strain segments randomly into training and test sets, ensuring that no training segment overlaps a test segment. Within the training and test sets, we generate both positive and negative examples. The negative examples contain only the strain data. For the positive examples, we inject waveforms into the noise segments by aligning the peak of the waveforms at the 90% location of the center 0.3 s, namely at the location of 0.42 s within the entire segment of 0.6 s. This choice was made as it safely covers the injected waveforms. The amplitude of the injection is set such that after filtering and whitening (to be described below), the resulting signal-to-noise ratio (SNR) is constant. For the experiment, the size of the training and test datasets are, respectively, 2.62 million and 2 million segments.

We preprocess all training and test data, by applying an FIR bandpass filter with cutoff frequencies 30 Hz and 400 Hz, whitening using a power spectral density estimated from the L1 strain data, and finally truncating to keep only the center 0.3 s (614 samples).

B. Matched filtering configuration

We first need to determine the necessary number of templates to use in matched filtering, given the space of parameters. We set 10, 100, 1000, and 10000 as the candidate numbers of templates. For each candidate number, we independently repeat the following process 30 times: randomly choose the specified number of pairs of parameters uniformly from $[40, 50] \times [40, 50]$, generate waveforms according to these parameters, preprocess (bandpass, whiten and truncate) as described above, and then normalize to equal power. This produces the templates for a matched filtering model. We evaluate the model on the test dataset to obtain an ROC curve. For each candidate number of templates and for each value of FPR, we take the lowest FNR outcome among the 30 independent runs. This is used to approximately represent the best performance achievable with a given number of templates.

The result is shown in Fig. 11. We see that the best performance of matched filtering in this setting starts to saturate at approximately 1000 templates, and the best performance with 1000 templates is almost identical to the that with 10000 templates. Therefore, we choose the best performance of matched filtering with 10000 templates, namely the bright blue curve, as the performance curve of the matched filtering method in this setting, against which we will be comparing our neural network method.

C. Neural network configuration

To initialize the templates of the neural network models for both `MNet-Shallow` and `MNet-Deep`, we generate 1000 random waveforms from a uniform distribution over

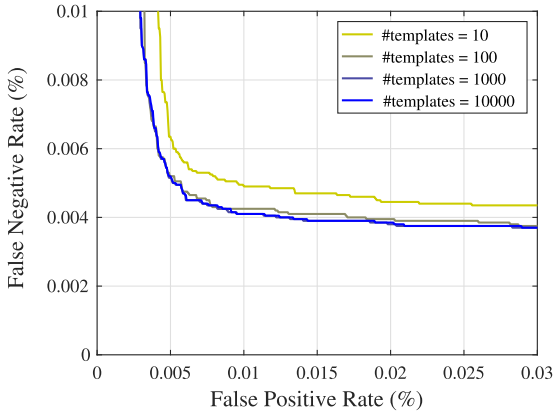

FIG. 11. The best performance of matched filtering with given number of templates across 30 independent runs. The performance starts to saturate above 1000 templates.

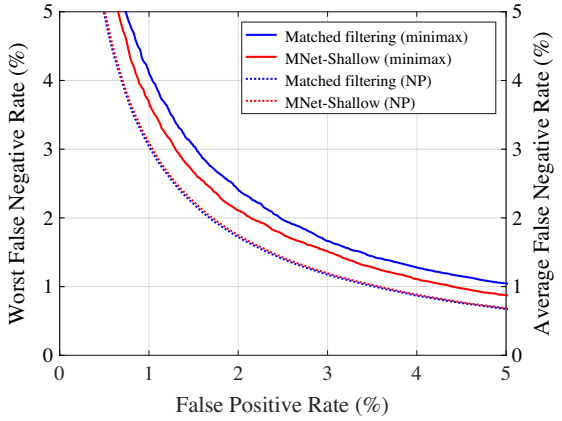

FIG. 12. ROC curves of the trained shallow neural network and matched filtering. The solid curves correspond to the vertical axis on the left, and the dotted curves correspond to the vertical axis on the right. For both models, we show both the worst (minimax) performance and the average performance under Neyman-Pearson (NP) setting with a uniform prior. The neural network with minimax training outperforms matched filtering in terms of the minimax criterion. The performance of the two models under NP is similar, which is reasonable since our optimization for the neural network was aimed for the minimax criterion only.

the same parameter range, subject to the same preprocessing and normalization process as done in matched filtering.

For the `MNet-Deep` architecture, in addition to the 1000 initialized templates, we also need to specify the number of layers and the feature dimension of each layer. In the experiment, we choose $L = 17$ and

$$(n_1, n_2, \ldots, n_L) = (1000, 1800, 1200, 720, 480, 300, 180, 120, 90, 60, 36, 24, 18, 12, 6, 3, 1).$$

Here, these feature dimensions n_l are chosen arbitrarily so long as they satisfy $n_2 \geq \frac{3}{2}n_1$, $n_\ell \geq \frac{1}{2}n_{l-1}$ for all $3 \leq \ell \leq L-1$, $n_{L-1} = 3$, $n_L = 1$, and that $n_2, \ldots, n_{L-2}$ are all divisible by 6 (which facilitates construction using our proposed initialization scheme).

For minimax training, in order to search the parameter space for the worst performance, we sample the parameter space $[40, 50] \times [40, 50]$ of (m_1, m_2) using a square grid sampler with interval 0.5. After discarding equivalent samples due to the symmetry between m_1 and m_2, there are in total 231 samples in the parameter space.

For the optimization parameters of the neural network, we train the network using logistic loss, the Adam optimizer [121], and a constant learning rate of 10^{-5}.

D. Simulation results

Performance under minimax. In this experiment, we perform injections such that SNR is 5, and only for the `MNet-Shallow` model. While this SNR value is smaller than the range of meaningful observed events, we choose this value for the simplicity of exposition and reduction of training time, since the training procedure for minimax criterion is rather computationally heavy. Similar results should hold at higher SNR values. Figure 12 plots the ROC curves for both matched filtering and `MNet-Shallow` trained for minimax, measured in terms of both worst performance and the average performance over a uniform prior. We see that the trained neural network achieves better performance than matched filtering under minimax, while achieving approximately identical performance as matched filtering under Neyman-Pearson with a uniform prior. This is not surprising since the training process is designed to only optimize for the minimax criterion, and not the Neyman-Pearson criterion with uniform prior.

Performance under a uniform prior. In this experiment, we perform injections such that SNR is 9. Figure 13 plots the ROC curves for both formulations `MNet-Shallow` and `MNet-Deep` trained for Neyman-Pearson, as well as that of matched filtering. As expected, the neural network

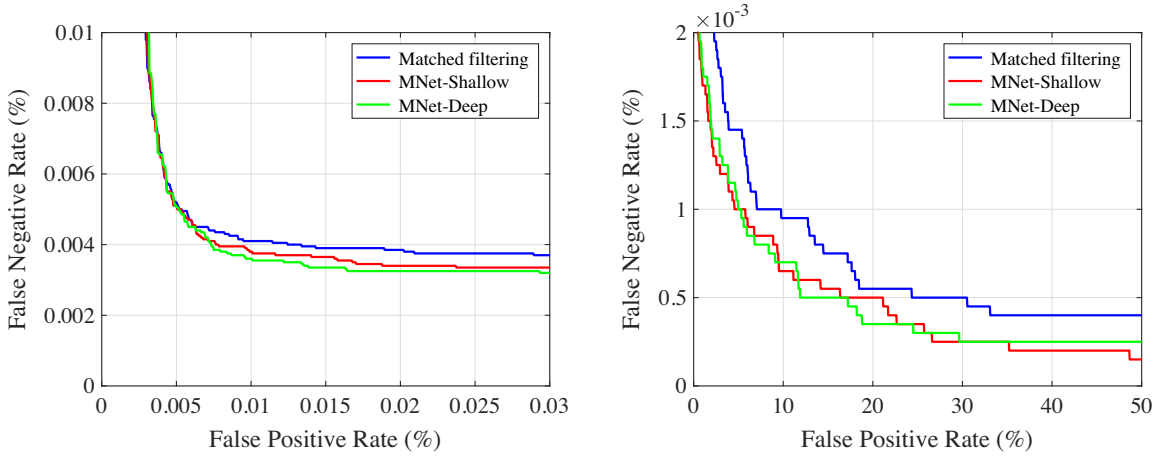

FIG. 13. ROC curves of the trained `MNet-Shallow` and `MNet-Deep` models compared with matched filtering. Left and right panels plot the same curves, but have different axis ranges to better show the contrast between the curves.

models strictly improves over matched filtering. Moreover, the MNet-Deep architecture has a slight performance advantage over MNet-Shallow. The performance improvement of the trained models over matched filtering is especially remarkable with low FNR values, which is arguably the more important scenario for gravitational wave detection, since we can hardly afford to miss actual astrophysical events which are quite scarce.

VII. DISCUSSION

Our experiments demonstrate the potential of neural networks to outperform matched filtering, especially at low false negative rates. The flexibility of neural networks also enables this architecture to implement more general variations of matched filtering, such as with weights or aggregation functions different from the maximum. Neural networks have additional potential advantages: deep networks can adapt to unknown and/or non-Gaussian noise distributions. In addition, architectural ideas in deep networks such as pooling help to convey invariances that may be helpful in detecting some "unknown unknowns" that lie outside of the span of a prespecified family of templates. This should be investigated in the future.

The proposed architectures can be adapted to time-varying noise distributions, by pretraining on very large collections of (synthetic) Gaussian noise and then adapting the pretrained network using a smaller number of online examples. This kind of pretraining may also be helpful in deploying our methods across larger mass ranges, which require more training data.

We note that it is, in some sense, unsurprising that deep networks can exhibit advantages over matched filtering, since the former can be made arbitrarily complex and can approximate essentially arbitrary functions. An important direction for future work is to study architectures that not only approach optimal statistical performance, but exhibit good *complexity-performance* tradeoffs. There are a number of concrete directions for achieving this—in particular, the weight matrices learned by our Neyman-Pearson networks exhibit particular types of low-dimensional (low-rank and sparse) structure, which can be leveraged to reduce complexity. Interpreting matched filtering as a particular neural network facilitates the study of complexity-performance tradeoffs, since it allows these distinct methods to be studied in a unified framework. Another avenue for complexity reduction is to define and train very large (overparametrized) networks and then prune them to produce much smaller subnetworks with good performance. MNet-Deep is particularly promising in this regard, since this construction yields networks of arbitrary depth.

One future possibility of the approach is to go beyond the fixed template banks that constrain the limited set of parameters taken into account. For example, to limit the size of the template bank, BH spins that are misaligned from the orbital angular momentum are not widely used yet. Also, due to the lack of available template banks, some astrophysically feasible scenarios receive relatively little attention, including eccentric binary merger template banks where every new template requires a computationally very expensive general-relativity simulation. Therefore, generalized matched filtering needs to be investigated in this context, to measure its performance on signal classes that current templates do not cover. Additionally, training it with a sample of eccentric waveforms could enable the detection of other eccentric BBHs even with properties not covered by the limited simulation used for training. Exploring these scenarios are very important experiments for the future.

Another desirable goal is to allow matched filtering algorithms to run "coherently," treating the GW detectors worldwide as a single detector and analyzing data from multiple GW detectors together as a single data stream. The main difficulty is that the sky direction of the cosmic source is unknown; therefore, there are many unknown time shifts among the detectors' data. Searching a large number of different combinations can be cost prohibitive with current approaches. It is important to experimentally investigate the ML extensions to matched filtering to measure the increased sensitivity due to the coherent framework.

Furthermore, experiments on the natural generalization of the approach where one does not aim to find the best matching waveform, but instead aims to estimate the parameters of the BBH system are needed. For example, instead of having the maximum reported, one could report the probability distribution over parameters. The difficulty here is that searches usually have much fewer parameters than what is used for parameter estimation. The performance of the ML framework in parameter estimation should be quantified in the future, even if it comes at the price of precision and is therefore only used as a first estimate.

VIII. CONCLUSION

In this paper, we highlighted the idea that matched filtering currently applied by LIGO is formally equivalent to a particular neural network, which can be defined analytically in closed form. We also modeled the LIGO gravitational wave search as the parametric signal detection problem and illustrated the suboptimality of matched filtering regardless of whether a prior distribution on the parameter space is given. On the other hand, we proposed neural network architectures MNet-Shallow and MNet-Deep, which are initialized to implement matched filtering exactly, and then trained on data for improved performance. In particular, we showed that when the prior distribution is known, the training process is aligned with the statistically optimal decision rule. Between the two proposed architectures, the former more closely resembles the architecture of matched filtering, while the latter has a more flexible

architecture capable of dealing with a wider range of distributions. We conducted experiments using LIGO strain data from O2 and synthetic waveform injections, and showed that our trained network can achieve uniformly better performance than matched filtering both with or without a known prior, especially in scenarios where false negative rate is low.

Through this work, we seek to bridge the gap between data-driven methods such as deep learning and those detection methods currently in use in LIGO, and explore the possibility of incorporating them into the gravitational wave search of LIGO, as well as broader areas of scientific discovery. In the future work, we aim to explore the potentials of efficiency gains of neural networks over matched filtering, and also establish an end-to-end guarantee for the performance of the proposed framework.

ACKNOWLEDGMENTS

We acknowledge computing resources from Columbia University's Shared Research Computing Facility project, which is supported by NIH Research Facility Improvement Grant No. 1G20RR030893-01, and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract No. C090171. The authors are grateful for the LIGO Scientific Collaboration for the careful review of the paper. This paper is assigned a LIGO DCC number of No. LIGO-P2100086. The authors acknowledge the LIGO Laboratory and Scientific Collaboration for the detectors, data, and the game changing computing resources (National Science Foundation Grants No. PHY-0757058 and No. PHY-0823459). The authors would like to thank colleagues of the LIGO Scientific Collaboration and the Virgo Collaboration and Columbia University for their help and useful comments, in particular the CBC group, Andrew Williamson, Stefan Countryman, William Tse, Nicolas Beltran, Asif Mallik, Sireesh Gururaja, and Thomas Dent which we hereby gratefully acknowledge. SM thanks David Spergel, Rainer Weiss, Rana Adhikari, and Kipp Canon for the motivating general discussions related to the role of machine learning and data analysis. This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center [95,122], a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain. The authors thank the University of Florida and Columbia University in the City of New York for their generous support.The authors are grateful for the generous support of the National Science Foundation under Grant No. CCF-1740391. The authors thank Sharon Sputz of Columbia University for her effort in facilitating this collaboration. I. B. acknowledges the support of the National Science Foundation under Grant No. PHY-1911796 and the Alfred P. Sloan Foundation.

APPENDIX: PROOFS OF KEY TECHNICAL CLAIMS

1. Proof of Proposition 1

Combining the definitions of the likelihood ratio $\lambda(\boldsymbol{x})$ and the probability densities $\rho_0(\boldsymbol{x})$ and $\rho_1(\boldsymbol{x})$, we have

$$\lambda(\boldsymbol{x}) = \frac{\int \rho_{\text{noise}}(\boldsymbol{x}-\boldsymbol{s}_{\boldsymbol{\gamma}})\mathrm{d}\nu(\boldsymbol{\gamma})}{\rho_{\text{noise}}(\boldsymbol{x})} \tag{A1}$$

$$= \int \frac{\rho_{\text{noise}}(\boldsymbol{x}-\boldsymbol{s}_{\boldsymbol{\gamma}})}{\rho_{\text{noise}}(\boldsymbol{x})}\mathrm{d}\nu(\boldsymbol{\gamma}). \tag{A2}$$

When the noise is Gaussian $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, the integrand equals

$$\frac{\rho_{\text{noise}}(\boldsymbol{x}-\boldsymbol{s}_{\boldsymbol{\gamma}})}{\rho_{\text{noise}}(\boldsymbol{x})} = \exp\left(\frac{\langle \boldsymbol{x}, \boldsymbol{s}_{\boldsymbol{\gamma}}\rangle - \|\boldsymbol{s}_{\boldsymbol{\gamma}}\|^2/2}{\sigma^2}\right), \tag{A3}$$

which is a convex function of $\boldsymbol{x}$. Hence after integrating over $\boldsymbol{\gamma}$, the resulting function $\lambda(\boldsymbol{x})$ is still a convex function of $\boldsymbol{x}$. The optimal decision region is a sublevel set of $\lambda(\boldsymbol{x})$ and is hence a convex set.

2. Proof of Proposition 2

Assume the training data is drawn iid from some distribution on $(\boldsymbol{x}, y) \in \mathbb{R}^n \times \{0, 1\}$. In this setting, the previous defined densities $p_0(\boldsymbol{x})$ and $p_1(\boldsymbol{x})$ can be expressed as $p_0(\boldsymbol{x}) = p(\boldsymbol{x}|y=0)$ and $p_1(\boldsymbol{x}) = p(\boldsymbol{x}|y=1)$. If the predictor function is $f: \mathbb{R}^n \to \mathbb{R}$, then the risk is

$$\mathcal{R}(f) = \mathbb{E}_{(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)] \tag{A4}$$

$$= \mathbb{P}[y=0]\cdot\mathbb{E}_{\boldsymbol{x}|y=0}[\ell(f(\boldsymbol{x}),0)] + \mathbb{P}[y=1]\cdot\mathbb{E}_{\boldsymbol{x}|y=1}[\ell(f(\boldsymbol{x}),1)] \tag{A5}$$

$$= \mathbb{P}[y=0]\int_{\mathbb{R}^n}\ell(f(\boldsymbol{x}),0)p_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \mathbb{P}[y=1]\int_{\mathbb{R}^n}\ell(f(\boldsymbol{x}),1)p_1(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{A6}$$

$$= \int_{\mathbb{R}^n} ((1-c)\ell(f(\boldsymbol{x}),0)p_0(\boldsymbol{x}) + c\ell(f(\boldsymbol{x}),1)p_1(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}, \tag{A7}$$

where $c \coloneqq \mathbb{P}[y=1] \in (0,1)$ is an exogenous constant that only depends on the data distribution. The function that minimizes the above risk is

$$f_\star(\boldsymbol{x}) = \arg\min_{\hat{y}} (1-c)\ell(\hat{y},0)p_0(\boldsymbol{x}) + c\ell(\hat{y},1)p_1(\boldsymbol{x}), \tag{A8}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$, or equivalently,

$$f_\star(\boldsymbol{x}) = \arg\min_{\hat{y}} \ell(\hat{y},0) + \frac{c\lambda(\boldsymbol{x})}{1-c}\ell(\hat{y},1). \tag{A9}$$

Therefore, the optimal predicted value at a point is the solution to an optimization problem that only depends on the likelihood ratio $\lambda(\boldsymbol{x})$.

Take an arbitrary fixed $\boldsymbol{x}$. From the assumption that $\ell(\hat{y},y)$ is strictly convex and minimized at $\hat{y}=y$, it follows that $\ell(\hat{y},0) + \frac{c\lambda(\boldsymbol{x})}{1-c}\ell(\hat{y},1)$ is strictly convex in $\hat{y}$, strictly decreasing on $(-\infty,0]$ and strictly increasing on $[1,\infty)$. Hence, for any $\boldsymbol{x}$, the risk minimization problem of Eq. (A9) has a unique solution in [0, 1]. The optimal solution can be found from the first-order-condition (FOC). Noticing that $\hat{y}$ cannot be 0 or 1 under the FOC, we can rewrite the FOC as

$$\frac{\ell'(\hat{y},0)}{-\ell'(\hat{y},1)} = \frac{c\lambda(\boldsymbol{x})}{1-c}. \tag{A10}$$

From the assumption of strong convexity, we know that on the interval (0,1) we have $\ell'(\hat{y},0) > 0$ and $\ell'(\hat{y},1) < 0$, where in ℓ' the derivative is taken with respect to the first argument. Hence the left-hand side of (A10) is strictly increasing in $\hat{y}$.

This concludes that the optimal decision function $f_\star(\boldsymbol{x})$ is strictly increasing in $\lambda(x)$.

[1] LIGO and Virgo Collaborations, Phys. Rev. Lett. **116**, 061102 (2016).
[2] B. P. Abbott *et al.*, Phys. Rev. X **9**, 031040 (2019).
[3] V. LSC and KAGRA Collaborations, Phys. Rev. X **11**, 021053 (2021).
[4] B. P. Abbott *et al.*, Phys. Rev. Lett. **119**, 161101 (2017).
[5] B. P. Abbott *et al.*, Astrophys. J. Lett. **848**, L12 (2017).
[6] T. Akutsu *et al.*, Prog. Theor. Exp. Phys. **2021**, 05A101 (2021).
[7] K. L. Dooley *et al.*, Classical Quantum Gravity **33**, 075009 (2016).
[8] F. Acernese *et al.*, Classical Quantum Gravity **32**, 024001 (2015).
[9] A. Abramovici, W. E. Althouse, R. W. P. Drever, Y. Gursel, S. Kawamura, F. J. Raab, D. Shoemaker, L. Sievers, R. E. Spero, K. S. Thorne, R. E. Vogt, R. Weiss, S. E. Whitcomb, and M. E. Zucker, Science **256**, 325 (1992).
[10] B. P. Abbott *et al.*, Classical Quantum Gravity **32**, 074001 (2015).
[11] C. Affeldt *et al.*, Classical Quantum Gravity **31**, 224002 (2014).
[12] F. Acernese *et al.* (Virgo Collaboration), Phys. Rev. Lett. **123**, 231108 (2019).
[13] M. Tse *et al.*, Phys. Rev. Lett. **123**, 231107 (2019).
[14] A. Einstein, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften, Berlin, 688 (1916).
[15] A. Einstein, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften, Berlin, 154 (1918).
[16] W. G. Anderson, P. R. Brady, J. D. Creighton, and É. É. Flanagan, Phys. Rev. D **63**, 042003 (2001).
[17] S. Klimenko and G. Mitselmakher, Classical Quantum Gravity **21**, S1819 (2004).
[18] W. G. Anderson, P. R. Brady, J. D. E. Creighton, and É. É. Flanagan, Int. J. Mod. Phys. D **09**, 303 (2000).
[19] S. W. Hawking and W. Israel, *Three Hundred Years of Gravitation* (Cambridge University Press, Cambridge, England, 1989).
[20] B. J. Owen and B. S. Sathyaprakash, Phys. Rev. D **60**, 022002 (1999).
[21] C. Cutler, T. A. Apostolatos, L. Bildsten, L. S. Finn, E. E. Flanagan, D. Kennefick, D. M. Markovic, A. Ori, E. Poisson, G. J. Sussman, and K. S. Thorne, Phys. Rev. Lett. **70**, 2984 (1993).
[22] C. Cutler and É. E. Flanagan, Phys. Rev. D **49**, 2658 (1994).
[23] É. É. Flanagan and S. A. Hughes, Phys. Rev. D **57**, 4535 (1998).
[24] É. É. Flanagan and S. A. Hughes, Phys. Rev. D **57**, 4566 (1998).
[25] B. e. a. Abbott, Phys. Rev. D **69**, 122001 (2004).
[26] Pycbc software releases, https://github.com/gwastro/pycbc/releases.
[27] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, Phys. Rev. D **85**, 122006 (2012).
[28] C. M. Biwer, C. D. Capano, S. De, M. Cabero, D. A. Brown, A. H. Nitz, and V. Raymond, Publ. Astron. Soc. Pac. **131**, 024503 (2019).
[29] B. Allen, Phys. Rev. D **71**, 062001 (2005).
[30] T. Dal Canton, A. H. Nitz, A. P. Lundgren, A. B. Nielsen, D. A. Brown, T. Dent, I. W. Harry, B. Krishnan, A. J. Miller, K. Wette, K. Wiesner, and J. L. Willis, Phys. Rev. D **90**, 082004 (2014).
[31] S. A. Usman *et al.*, Classical Quantum Gravity **33**, 215004 (2016).

[32] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, Phys. Rev. D **98**, 024050 (2018).
[33] C. Messick *et al.*, Phys. Rev. D **95**, 042001 (2017).
[34] S. Sachdev *et al.*, arXiv:1901.08580.
[35] C. Hanna, S. Caudill, C. Messick, A. Reza, S. Sachdev, L. Tsukada, K. Cannon, K. Blackburn, J. D. E. Creighton, H. Fong, P. Godwin, S. Kapadia, T. G. F. Li, R. Magee, D. Meacher, D. Mukherjee, A. Pace, S. Privitera, R. K. L. Lo, and L. Wade, Phys. Rev. D **101**, 022003 (2020).
[36] Q. Chu, Low-latency detection and localization of gravitational waves from compact binary coalescences, Ph.D. thesis, The University of Western Australia, 2017.
[37] T. Adams, D. Buskulic, V. Germain, G. M. Guidi, F. Marion, M. Montani, B. Mours, F. Piergiovanni, and G. Wang, Classical Quantum Gravity **33**, 175012 (2016).
[38] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, Astrophys. J. **849**, 118 (2017).
[39] A. C. SearlearXiv:0804.1161.
[40] R. Biswas, P. R. Brady, J. Burguet-Castell, K. Cannon, J. Clayton, A. Dietz, N. Fotopoulos, L. M. Goggin, D. Keppel, C. Pankow, L. R. Price, and R. Vaulin, Phys. Rev. D **85**, 122008 (2012).
[41] T. Dent and J. Veitch, Phys. Rev. D **89**, 062002 (2014).
[42] B. J. Owen and B. S. Sathyaprakash, Phys. Rev. D **60**, 022002 (1999).
[43] B. J. Owen, Phys. Rev. D **53**, 6749 (1996).
[44] B. S. Sathyaprakash and S. V. Dhurandhar, Phys. Rev. D **44**, 3819 (1991).
[45] S. V. Dhurandhar and B. S. Sathyaprakash, Phys. Rev. D **49**, 1707 (1994).
[46] B. P. Abbott *et al.*, Classical Quantum Gravity **33**, 134001 (2016).
[47] B. P. Abbott *et al.*, Classical Quantum Gravity **37**, 055002 (2020).
[48] V. Gayathri, J. Healy, J. Lange, B. O'Brien, M. Szczepanczyk, I. Bartos, M. Campanelli, S. Klimenko, C. Lousto, and R. O'Shaughnessy, arXiv:2009.05461.
[49] T. Gebhard, N. Kilbertus, G. Parascandolo, I. Harry, and B. Schölkopf, in *Workshop on Deep Learning for Physical Sciences (DLPS) at the 31st Conference on Neural Information Processing Systems (NIPS)* (2017), pp. 1–6, https://dl4physicalsciences.github.io.
[50] D. George and E. Huerta, Phys. Lett. B **778**, 64 (2018).
[51] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, Phys. Rev. Lett. **120**, 141103 (2018).
[52] D. George and E. Huerta, Phys. Rev. D **97**, 044039 (2018).
[53] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, Sci. China: Phys., Mech. Astron. **62**, 969512 (2019).
[54] F. Morawski, M. Bejger, and P. Ciecielag, Mach. Learn. Sci. Technol. **1**, 025016 (2020).
[55] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, Phys. Rev. D **100**, 044009 (2019).
[56] P. G. Krastev, Phys. Lett. B **803**, 135330 (2020).
[57] B.-J. Lin, X.-R. Li, and W.-L. Yu, Front. Phys. **15**, 1 (2020).
[58] Y.-C. Lin and J.-H. P. Wu, Phys. Rev. D **103**, 063034 (2021).
[59] C. Bresten and J.-H. Jung, arXiv:1910.08245.
[60] P. Astone, P. Cerdá-Durán, I. Di Palma, M. Drago, F. Muciaccia, C. Palomba, and F. Ricci, Phys. Rev. D **98**, 122002 (2018).
[61] T. S. Yamamoto and T. Tanaka, arXiv:2011.12522.
[62] C. Dreissigacker and R. Prix, Phys. Rev. D **102**, 022005 (2020).
[63] R. Corizzo, M. Ceci, E. Zdravevski, and N. Japkowicz, Expert Systems Appl. **151**, 113378 (2020).
[64] A. L. Miller, P. Astone, S. DAntonio, S. Frasca, G. Intini, I. La Rosa, P. Leaci, S. Mastrogiovanni, F. Muciaccia, A. Mitidis *et al.*, Phys. Rev. D **100**, 062005 (2019).
[65] J. Bayley, C. Messenger, and G. Woan, Phys. Rev. D **102**, 083024 (2020).
[66] P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, arXiv:2012.13101.
[67] H.-M. Luo, W. Lin, Z.-C. Chen, and Q.-G. Huang, Front. Phys. **15**, 1 (2020).
[68] G. R. Santos, M. P. Figueiredo, A. d. P. Santos, P. Protopapas, and T. A. Ferreira, arXiv:2003.09995.
[69] M. L. Chan, I. S. Heng, and C. Messenger, Phys. Rev. D **102**, 043022 (2020).
[70] H. Xia, L. Shao, J. Zhao, and Z. Cao, Phys. Rev. D **103**, 024040 (2021).
[71] F. Acernese *et al.*, Classical Quantum Gravity **32**, 024001 (2015).
[72] LIGO Scientific Collaboration, Classical Quantum Gravity **32**, 074001 (2015).
[73] T. Akutsu *et al.*, Prog. Theor. Exp. Phys. **2021**, 05A101 (2021).
[74] R. Biswas, L. Blackburn, J. Cao, R. Essick, K. A. Hodge, E. Katsavounidis, K. Kim, Y.-M. Kim, E.-O. Le Bigot, C.-H. Lee *et al.*, Phys. Rev. D **88**, 062003 (2013).
[75] D. George, H. Shen, and E. Huerta, arXiv:1706.07446.
[76] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, Phys. Rev. D **95**, 104059 (2017).
[77] M. Razzano and E. Cuoco, Classical Quantum Gravity **35**, 095016 (2018).
[78] S. Coughlin, S. Bahaadini, N. Rohani, M. Zevin, O. Patane, M. Harandi, C. Jackson, V. Noroozi, S. Allen, J. Areeda *et al.*, Phys. Rev. D **99**, 082002 (2019).
[79] R. E. Colgan, K. R. Corley, Y. Lau, I. Bartos, J. N. Wright, Z. Márka, and S. Márka, Phys. Rev. D **101**, 102003 (2020).
[80] H. Nakano, T. Narikawa, K.-I. Oohara, K. Sakai, H.-A. Shinkai, H. Takahashi, T. Tanaka, N. Uchikata, S. Yamamoto, and T. S. Yamamoto, Phys. Rev. D **99**, 124032 (2019).
[81] S. R. Green, C. Simpson, and J. Gair, Phys. Rev. D **102**, 104057 (2020).
[82] J. P. Marulanda, C. Santa, and A. E. Romano, Phys. Lett. B **810**, 135790 (2020).
[83] A. Caramete, A. Constantinescu, L. Caramete, T. Popescu, R. Balasov, D. Felea, M. Rusu, P. Stefanescu, and O. Tintareanu, arXiv:2009.06109.
[84] A. Delaunoy, A. Wehenkel, T. Hinderer, S. Nissanke, C. Weniger, A. R. Williamson, and G. Louppe, arXiv:2010.12931.
[85] H. Shen, D. George, E. A. Huerta, and Z. Zhao, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York, 2019), pp. 3237–3241.
[86] W. Wei and E. Huerta, Phys. Lett. B **800**, 135081 (2020).
[87] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, Phys. Rev. D **100**, 063015 (2019).

[88] R. Balestriero and R. Baraniuk, arXiv:1805.06576.
[89] J. Cheng, Y. Wu, W. AbdAlmageed, and P. Natarajan, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 11553–11562, https://openaccess.thecvf.com/content_CVPR_2019/html/Cheng_QATM_Quality-Aware_Template_Matching_for_Deep_Learning_CVPR_2019_paper.html.
[90] J. M. Bower, Y.-F. Wong, and J. Banik, in *Neural Information Processing Systems* (1988), pp. 103–113.
[91] D. Tank and J. Hopfield, IEEE Trans. Circuits Systems **33**, 533 (1986).
[92] D. Buniatyan, T. Macrina, D. Ih, J. Zung, and H. S. Seung, arXiv:1705.08593.
[93] Q. Xue, Y. H. Hu, and W. J. Tompkins, IEEE Trans. Biomed. Eng. **39**, 317 (1992).
[94] R. P. Lippmann and P. Beckman, in *Advances in Neural Information Processing Systems* (1989), pp. 124–132.
[95] R. Abbott *et al.*, SoftwareX **13**, 100658 (2021).
[96] L. S. Finn, Phys. Rev. D **46**, 5236 (1992).
[97] In general, a global Earth and Space based gravitational-wave detector network can be treated as a composite data stream [42,98]. However, that added complexity is not necessary when discussing the principles of the paper. Therefore, we constrain ourselves to a single datastream in this proof of principle analysis.
[98] L. S. Finn, Phys. Rev. D **63**, 102001 (2001).
[99] G. Casella and R. L. Berger, *Statistical Inference* (Cengage Learning, Boston, USA, 2021).
[100] Q. Yu, The Can. J. Stat./La Revue Canadienne de Statistique 281 (1992).
[101] Y. LeCun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).
[102] D. Scherer, A. Müller, and S. Behnke, in *International Conference on Artificial Neural Networks* (Springer, New York, 2010), pp. 92–101.
[103] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning* (MIT Press Massachusetts, USA, 2017), Vol. 1.
[104] G. Cybenko, Math. Control, Signals Systems **2**, 303 (1989).
[105] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, Neural Netw. **6**, 861 (1993).
[106] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, Adv. Neural Inf. Process. Syst. **30**, 6231 (2017).
[107] S. Buchanan, D. Gilboa, and J. Wright, arXiv:2008.11245.
[108] K. Gregor and Y. LeCun, in *Proceedings of the 27th International Conference on Machine Learning* (2010), pp. 399–406.
[109] J. Bruna and S. Mallat, IEEE Trans. Pattern Analysis Mach. Intell. **35**, 1872 (2013).
[110] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 370–378.
[111] J. Sun, H. Li, Z. Xu *et al.*, in Adv. Neural Inf. Process. Syst. (2016), pp. 10–18.
[112] We note that the representation is not unique, and can be subject to shift and scale to produce essentially the same decision rule. Specifically, $\boldsymbol{b}^1$ can be identity vector times a constant (including zero) and $\boldsymbol{W}^1$ can be scaled by an arbitrary positive constant. However we choose the form given here for simplicity.
[113] P. Kidger and T. Lyons, in *Conference on Learning Theory* (PMLR, 2020), pp. 2306–2327.
[114] P. Orponen *et al.*, Nordic Journal of Computing **1994**, 94 (1994).
[115] M. Bianchini and F. Scarselli, IEEE Trans. Neural Networks Learn. Systems **25**, 1553 (2014).
[116] In fact it is straightforward to show that the conclusion of Proposition 2 holds for more general classes of loss functions, including the logistic loss.
[117] J. Creighton and W. Anderson, *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis* (John Wiley & Sons, Hoboken, USA, 2011).
[118] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar, Phys. Rev. D **53**, 3033 (1996).
[119] P. R. Brady, T. Creighton, C. Cutler, and B. F. Schutz, Phys. Rev. D **57**, 2101 (1998).
[120] C. Messenger, R. Prix, and M. A. Papa, Phys. Rev. D **79**, 104017 (2009).
[121] D. P. Kingma and J. Ba, arXiv:1412.6980.
[122] GWOSC, Gravitational wave open science center, https://www.gw-openscience.org/.