Interpretable Machine Learning Architectures for Efficient Signal Detection with Applications to Gravitational Wave Astronomy

Jingkai Yan

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2024

Abstract

Interpretable Machine Learning Architectures for Efficient Signal Detection
with Applications to Gravitational Wave Astronomy

Jingkai Yan

Deep learning has seen rapid evolution in the past decade, accomplishing tasks that were previously unimaginable. At the same time, researchers strive to better understand and interpret the underlying mechanisms of the deep models, which are often justifiably regarded as "black boxes". Overcoming this deficiency will not only serve to suggest better learning architectures and training methods, but also extend deep learning to scenarios where interpretability is key to the application. One such scenario is signal detection and estimation, with gravitational wave detection as a specific example, where classic methods are often preferred for their interpretability. Nonetheless, while classic statistical detection methods such as matched filtering excel in their simplicity and intuitiveness, they can be suboptimal in terms of both accuracy and computational efficiency. Therefore, it is appealing to have methods that achieve "the best of both worlds", namely enjoying simultaneously excellent performance and interpretability.

In this thesis, we aim to bridge this gap between modern deep learning and classic statistical detection, by revisiting the signal detection problem from a new perspective. First, to address the perceived distinction in interpretability between classic matched filtering and deep learning, we state the intrinsic connections between the two families of methods, and identify how trainable networks can address the structural limitations of matched filtering. Based on these ideas, we propose two trainable architectures that are constructed based on matched filtering, but

with learnable templates and adaptivity to unknown noise distributions, and therefore higher detection accuracy. We next turn our attention toward improving the computational efficiency of detection, where we aim to design architectures that leverage structures within the problem for efficiency gains. By leveraging the statistical structure of class imbalance, we integrate hierarchical detection into trainable networks, and use a novel loss function which explicitly encodes both detection accuracy and efficiency. Furthermore, by leveraging the geometric structure of the signal set, we consider using signal space optimization as an alternative computational primitive for detection, which is intuitively more efficient than covering with a template bank. We theoretical prove the efficiency gain by analyzing Riemannian gradient descent on the signal manifold, which reveals an exponential improvement in efficiency over matched filtering. We also propose a practical trainable architecture for template optimization, which makes use of signal embedding and kernel interpolation.

We demonstrate the performance of all proposed architectures on the task of gravitational wave detection in astrophysics, where matched filtering is the current method of choice. The architectures are also widely applicable to general signal or pattern detection tasks, which we exemplify with the handwritten digit recognition task using the template optimization architecture. Together, we hope the this work useful to scientists and engineers seeking machine learning architectures with high performance and interpretability, and contribute to our understanding of deep learning as a whole.

Table of Contents

List of Figures

Acknowledgements

Needless to say, this achievement of mine would not have been remotely possible without the great support I received from everyone around me along the way. Pardon my relative conciseness here, for no amount of words can fully convey such gratitude.

To my dear parents, who devoted so much of everything to me, and have been my eternal source of strength. This is something I can never repay.

To my advisor John, who guided me through the ups and downs of this academic journey, and taught me the ways of both research and life. Your knowledge and wisdom never fail to impress. To Szabi and Zsuzsa, who introduced me an amazing world of ideas, letting my mind soar in the universe. Also to Dan and Marianthi, who led me into various exciting domains of science.

To my senior labmates, Henry, Robert and Sam, who gave me plentiful guidance with your patience throughout the many projects. And to my peer labmates, Mariam and Tim, who walked this path with me, and gave me lots of inspiration along the way.

Finally, to my partner Shuhua, who has accompanied me throughout the entirety of what exists on these pages, and every moment of these years. Let us continue to write the future pages together.

Again, to all of you who have helped and supported me through this journey, thank you.

Dedication

To my family.

Chapter 1: Introduction

1.1 Deep Learning and Interpretable Architectures

Deep learning has been evolving at exceptional speed in the recent years. With the advent of new deep learning architectures, from convolutional networks [1, 2] and LSTMs [3] to transformers [4, 5] and diffusion models [6], deep learning is capable of accomplishing tasks that were unimaginable to people decades ago, such as image recognition [2, 5], language synthesis [4, 7], game playing [8, 9], and so on. These methods are often empowered by over-parameterized architectures and enormous training datasets. For example, the GPT-3 model [7] released in 2020 has 175 billion parameters which require 800GB to store.

Such impressive scaling is undoubtedly a major factor for the success of modern deep learning. At the same time, along with constructing ever-expanding modern architectures, researchers have been striving to better understand the underlying mechanisms, for both recent transformers and diffusion models [10, 11] and the more basic fully-connected networks alike [12, 13]. Although plenty of insight has been obtained in the literature, there remain far more questions than answers, and it is not surprising that deep learning methods are often regarded as "black boxes" [14]. Demystifying these black box models will not only help to better interpret the learned features and outputs, but also point toward better architectural designs, and potentially expand the scope of applicability of these models. Many existing works aim to address this interpretability issue of black box models from various angles, such as finding alternative transparent-box models that mimic the behavior of the black box [15, 16], or obtaining explanations for specific decision-making of the black box [17, 18, 19]. A more in-depth summary of the different approaches can be found in the survey paper [14].

Another approach to the interpretability of learning architectures is to instead use interpretable

methods as a starting point. Powerful as modern black box architectures are, they are still far from being almighty and there are places where traditional methods shine. For example, linear regression and its variants are still the predominant methods in economics [20, 21], largely due to their statistical explanatory power which is critical to economic applications. Medical imaging also depends heavily on traditional processing techniques, due to often limited dataset sizes and vaguely-defined problems [22]. These traditional methods are often well-motivated and intuitive, enjoying desirable efficiency and interpretability, but sometimes lacking in performance compared with modern deep learning methods which takes advantage of massive amounts of training data, provided that is available. This suggests a possibility of equipping traditional methods with trainable components, enabling it to leverage the power of data-driven methods while maintaining the strengths of model-based methods, effectively achieving "the best of both worlds". As a specific example, unrolled optimization [23, 24, 25] is a technique where an iterative optimization algorithm is reconfigured to become trainable, thereby improving the model performance over the original optimization method. Granted, such model-based interpretable architectures are not designed to replace modern transformers altogether, but rather suggesting an alternative approach to well-defined problem setups where computational efficiency and model interpretability are a key concern. One such scenario is the classic statistical problem of signal detection and estimation, which we introduce in the following section.

1.2 The Signal Detection Problem

The detection and estimation of signals from noise is a classic problem in statistical signal processing with well-founded theory [26], and has wide applications ranging from radar and geophysics to image processing and biomedical engineering. In this section, we will first state the problem formulation, and then briefly describe some classic and deep learning approaches to it.

Assume our signals of interest form a certain set $S \subset \mathbb{R}^D$, which is typically a d-dimensional submanifold of $\mathbb{R}^D$. Often the set S can be indexed by a parameterization, in which case we also refer to the problem as "parametric signal detection". Given a noisy observation $\boldsymbol{x} \in \mathbb{R}^D$, the dual

tasks of detection and estimation are to determine whether $\boldsymbol{x}$ contains some signal of interest or not, and find the corresponding signal if it exists. More formally, we model tuples of observations and labels $(\boldsymbol{x}, y) \in \mathbb{R}^D \times \{0, 1\}$ as:

$$\boldsymbol{x} = \begin{cases} a\,\boldsymbol{s}_\natural + \boldsymbol{z} & \text{if } y = 1 \\ \boldsymbol{z}, & \text{if } y = 0 \end{cases} \tag{1.1}$$

where $a \in \mathbb{R}_+$ is the signal amplitude, $\boldsymbol{s}_\natural \in S$ is the ground truth signal contained in $\boldsymbol{x}$, and $\boldsymbol{z}$ is additive noise. We assume the signals are normalized to have unit power, namely $\|\boldsymbol{s}\|^2 = 1$ for all $\boldsymbol{s} \in S$. For the simplicity of illustration, we assume the noise is white Gaussian, namely $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, and implications of non-Gaussian noise will be discussed later in the thesis. Given an input $\boldsymbol{x}$, we want to predict the corresponding label y, as well as the signal $\boldsymbol{s}_\natural$ if $y = 1$.

An aspect of the data model that remains unspecified is how the signal $\boldsymbol{s}_\natural$ is drawn from the signal set S. For the majority of the discussion, we will adopt the perspective that $\boldsymbol{s}_\natural$ is drawn from a certain signal distribution, and that the evaluation of model performance follows the same distribution. At the same time, we will also briefly discuss implications of a minimax perspective, where model performance is evaluated based on the worst performance over all possible $\boldsymbol{s}_\natural$ from the signal set.

When detecting a single target signal in additive noise, it is known that matched filtering (MF) gives the optimal linear filter for maximizing the signal-to-noise ratio (SNR). When detecting a family of signals, however, matched filtering is in general no longer optimal, but its simplicity and intuitiveness still have a strong appeal. The multi-signal variant of matched filtering uses the conceptual decision statistic $\max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle$.[1] In other words, it implements

$$\hat{y}(\boldsymbol{x}) = 1 \iff \max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle \geq \tau \tag{1.2}$$

where τ is a given threshold, and the estimated signal can be obtained as $\arg\max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle$.

[1] The general suboptimality of this statistic will be discussed later in the thesis.

Matched filtering, or template matching, approximates the above decision statistic with the maximum over a finite bank of templates $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K$ of size K:

$$\hat{y}_{\mathrm{MF}}(\boldsymbol{x}) = 1 \iff \max_{i=1,\ldots,K} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle \geq \tau. \tag{1.3}$$

The estimated signal is then the template $\boldsymbol{s}_i$ that contributes to the highest correlation. If the template bank densely covers S, (1.3) will accurately approximate (1.2). However, dense covering can be very inefficient — the number of templates required to cover S with some target radius r grows as $n \propto 1/r^d$, making this approach impractical for all but the smallest d. This inefficiency has motivated significant efforts in applied communities to optimize the placement of the templates $\boldsymbol{s}_i$, maximizing the statistical performance for a given fixed K [27]. Nevertheless, the curse of dimensionality remains in force. Note that our model is simplified in that the location of signal occurrence is assumed to be known, as indicated by the fixed input dimension. However, the distinction between unknown versus known location can be resolved by replacing the inner product operation with the correlation (or convolution) operation, and there is no loss assuming fixed location for matched filtering.

This conceptual idea of using large template banks for detection is widely present in applications such as neuroscience [28], geophysics [29, 30], image pose recognition [31], radar signal processing [32, 33], and aerospace engineering [34], amongst many others. In the meantime, many modern learning architectures employ similar ideas of matching inputs with template banks, such as transformation-invariant neural networks which create a large number of templates by applying transformations to a smaller family of filters [35, 36, 37].

Alongside matched filtering and other statistical signal methods, deep learning suggests a different approach to the problem. In the above problem setup (1.1) where the input has fixed dimension, a fully-connected feedforward neural network can output binary labels for signal detection, or output signal parameters for signal estimation. Again, the fully-connected network naturally takes the form of a convolutional network when the signal location is unknown, where the convolution

kernels are identical to the weights in the fully-connected network.

As previously discussed, both families of methods are viable and have different strengths. The classic statistical methods excel in their intuitiveness and interpretability, while the deep learning methods often enjoy better performance. Although deep learning methods are gaining popularity in many tasks, there remain scenarios where the interpretability is highly valued and therefore more traditional approach is preferred. In the following section, we introduce the task of gravitational wave detection from astrophysics, where despite plenty of deep learning architectures having been proposed in the literature, matched filtering remains the current method of choice.

1.3 A Specific Application: Gravitational Wave Detection

The discovery of cosmic gravitational waves [38], the windfall of binary black-hole merger detections [39, 40], and the spectacular insights that multimessenger astrophysics provided [41, 42] 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 [43], GEO600 [44], Virgo [45], and LIGO [46, 47], played a critical role as they provided the technology [48, 49, 50] enabling signals to be extracted from ripples in Einstein's space-time [51, 52]. 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 problem of identifying gravitational waves [53] in a single gravitational-wave detector data stream [2] can be formulated as follows: we observe detector strain data $\boldsymbol{x} \in \mathbb{R}^D$, and wish to determine whether $\boldsymbol{x}$ consists of astrophysical signal plus noise, or noise alone. Furthermore, the physical parameters associated with gravitational wave signals serve as natural parameters for the signals, and hence this aligns well with the parametric signal detection and estimation problem stated in the previous section.

The bulk of the discoveries were made by two classes of powerful data analysis approaches,

[2] In general, a global Earth and Space based gravitational-wave detector network can be treated as a composite data stream [54, 55]. However, that added complexity is unnecessary when discussing the principles of this work. Therefore, we constrain ourselves to a single datastream in this proof of principle analysis.

excess power [56, 57, 58] and matched filtering [59, 60, 61, 62, 63, 64, 65]. The flagship matched filtering methods [66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78] reached unprecedented sophistication and became the workhorse of the field [39, 40]. Insightful work also exist on the extent of optimality, role of intrinsic parameters, and effect of non-Gaussian backgrounds [79, 80, 81]. There is more than historical evidence on their algorithmic power [54], and they are also considered optimal [62] when searching for chirps of known shape [82, 60, 83, 84] embedded in well-behaved Gaussian noise. Within the optimality and success lie limitations, as the data is significantly more complex [85, 86] than Gaussian noise and many cosmic signals are not as well known as the binary black-hole models that are being used in searches [87]. 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.

At the same time, 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 performance approaching the matched filtering search currently used by LIGO [88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109]. In addition, these machine learning methods can also be applied to glitches and noise transients identification [110, 111, 112, 113, 92, 114, 115], signal classification and parameter estimation [116, 117, 118, 119, 120], data denoising [121, 122], 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 [123].

As we deepen our search for gravitational wave signals, the issue of computational efficiency (namely, the number of basic operations required by a computer) is becoming increasingly prominent. Detection methods that excel in both statistical performance and computational efficiency can significantly boost our capacities for exploring wider and higher-dimensional parameter spaces, and even other families of eccentric waveforms [124]. This in turn will help with uncovering more

astrophysical events, potentially unveiling novel astrophysical phenomena, as well as reducing the carbon footprint associated with searching for these events.

1.4 Objectives and Structure of the Thesis

Based on the existing body of literature, we aim to bridge the gap between traditional statistical methods and novel deep learning methods for signal detection, with particular applications to gravitational wave detection. In Chapter 2 [125], we first establish the intrinsic equivalence and connections between matched filtering and neural networks, and illustrate the structural limitations of matched filtering. We then propose two network architectures, shallow and deep respectively, which are specifically constructed to replicate matched filtering at initialization and then trainable on data, and are able to achieve better statistical accuracy than the baseline matched filtering. In Chapter 3 [126], we pivot around the computational efficiency of detection, and propose a hierarchical trainable architecture that uses multi-layer decision rules to take advantage of the detection problem structure. The model is trained using a loss function designed to explicitly encode both accuracy and efficiency, which are the two desiderata for the task. In Chapter 4, we utilize a different type of structure in the problem, namely the geometric structure of the signal set, and propose to use optimization to replace covering (as done in matched filtering) as the computational primitive. We show theoretically that optimization is exponentially more efficient than covering for signal detection. Furthermore, we apply unrolled optimization and kernel interpolation to the iterative method, and propose a trainable architecture that excels in both efficiency and interpretability. Finally, we provide discussions and concluding remarks in Chapter 5.

Chapter 2: Generalized Approach to Matched Filtering Using Neural Networks

2.1 Introduction

This work 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 [127, 128, 129, 130, 131, 132, 133]. 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 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 distribu-

tion, 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 [134] 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 chapter is organized as follows. Section 2.2 discusses two possible formulations of the objective. Section 2.3 discusses matched filtering as an approach to solving the parametric detection problem, as well as its limitations. Section 2.4 illustrates how neural network models can be applied in this problem, in a way that exactly implements matched filtering at initialization. Section 2.5 discusses the training process of neural network models, and in particular how it is aligned with the parametric signal detection problem. In Section 2.6 we present experimental results on real LIGO data and synthetic injections. We discuss some further implications of this work in Section 2.7, and conclude in Section 2.8.

2.2 Two Possible Objectives for Parametric Signal Detection

As discussed in the introductory chapter, the problem of detecting a parametric family of gravitational wave signals can be modeled as the following hypothesis test:

$$H_0 : \quad \boldsymbol{x} = \boldsymbol{z}, \tag{2.1}$$

$$\text{or} \qquad H_1 : \quad \boldsymbol{x} = \boldsymbol{s}_\natural + \boldsymbol{z} \ \text{ for some } \ \boldsymbol{s}_\natural \in S. \tag{2.2}$$

Our broad goal is to identify decision rules $\delta : \mathbb{R}^D \to \{0, 1\}$ 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 ground truth signal $\boldsymbol{s}_\natural$ is drawn from S with some known distribution ν, and the *minimax* framework, in which we control the worst performance over all possible cases of the ground truth signal $\boldsymbol{s}_\natural$.

2.2.1 Neyman-Pearson Framework

In this setting, one assumes a known probability distribution ν for the ground truth signal, which enables us to view H_1 as a simple hypothesis. The false positive rate (FPR) associated with the rule δ is

$$\mathrm{FPR} = \mathbb{P}_{\boldsymbol{z}} \left[\delta(\boldsymbol{z}) = 1\right] \tag{2.3}$$

The false negative rate (FNR) associated with a specific signal $\boldsymbol{s}$ is

$$\mathrm{FNR}(\boldsymbol{s}) = \mathbb{P}_{\boldsymbol{z}} \left[\delta\left(\boldsymbol{s} + \boldsymbol{z}\right) = 0\right]. \tag{2.4}$$

The *overall* false negative rate is

$$\mathrm{FNR} = \int \mathrm{FNR}(\boldsymbol{s}) \, \mathrm{d}\nu(\boldsymbol{s}). \tag{2.5}$$

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

$$\min_{\delta} \ \mathrm{FNR} \quad \text{subject to} \quad \mathrm{FPR} \leq \alpha, \tag{2.6}$$

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,

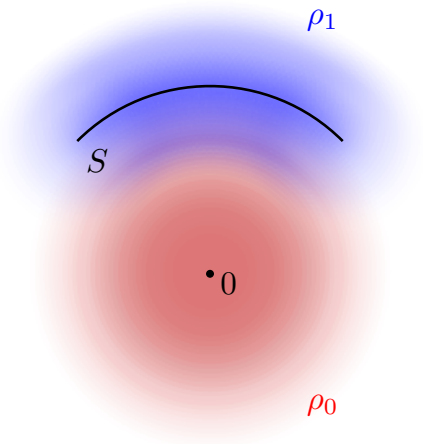

Figure 2.1: An example of the parametric signal detection problem with signal space S. Densities ρ_0 and ρ_1 are shown in red and blue respectively.

respectively, then the optimal test is given by comparing the *likelihood ratio*

$$\lambda(\boldsymbol{x}) = \frac{\rho_1(\boldsymbol{x})}{\rho_0(\boldsymbol{x})} \tag{2.7}$$

to a threshold τ, which depends on the significance level α. An illustration of an example problem is shown in FIG 2.1.

2.2.2 Minimax Framework

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

$$\min \ \text{WFNR} \quad \text{subject to} \quad \text{FPR} \leq \alpha. \tag{2.8}$$

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

$$\text{WFNR} \ = \ \max_{\boldsymbol{s} \in S} \text{FNR}(\boldsymbol{s}). \tag{2.9}$$

In contrast to the Neyman-Pearson criterion, there is in general no simple expression for the minimax optimal rule δ [135]. In the next section, we will review matched filtering, a simple, popular approach to detection which is compatible with the minimax framework (albeit suboptimal in terms of (2.8)), in the sense that it does not require a prior on the signal distribution.

2.3 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).

2.3.1 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., $\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^*\right] = \sigma^2\boldsymbol{I}$), matched filtering simply computes the inner product between the target $\boldsymbol{s}$ and the observation:

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \langle \boldsymbol{s}, \boldsymbol{x} \rangle \geq \tau. \tag{2.10}$$

When detecting a single signal $\boldsymbol{s}$ in iid Gaussian noise, this decision rule is optimal in both the Neyman-Pearson and minimax senses: for example, if $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, the likelihood ratio

$$\lambda(\boldsymbol{x}) = \frac{\rho_0(\boldsymbol{x} - \boldsymbol{s})}{\rho_0(\boldsymbol{x})} = \exp\left(\frac{\langle \boldsymbol{s}, \boldsymbol{x} \rangle - \|\boldsymbol{s}\|^2/2}{\sigma^2}\right) \tag{2.11}$$

is a monotone function of $\langle \boldsymbol{s}, \boldsymbol{x} \rangle$, and so matched filtering implements the (optimal) likelihood ratio test. FIG 2.2 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 is suboptimal in both the Neyman-Pearson and minimax settings. Moreover, there are a number of additional factors which contribute to its suboptimality. These in-

clude 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 [62]. Nevertheless, we will see how matched filtering can inspire principled approaches to deriving more flexible decision rules which can address many of these challenges.

2.3.2 Extensions to Parametric Detection

The simplest extension of the decision rule (2.10) to *parametric* detection problems, in which there are multiple potential targets signals, involves taking the maximum over the parameter space:

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \max_{\boldsymbol{s}\in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle \geq \tau. \tag{2.12}$$

Here we used the assumption that all templates have unit norm, namely $\|\boldsymbol{s}\|_2^2 = 1,\ \forall \boldsymbol{s} \in S$. When this rule (2.12) is hard to implement in exact form, it can typically be approximated by taking samples $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K$ and setting

$$\delta(\boldsymbol{x}) = 1 \text{ iff } \max_{i=1,\ldots,K} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle \geq \tau. \tag{2.13}$$

When the sampling is sufficiently dense, the sampled matched filter rule (2.13) accurately approximates the ideal matched filter rule (2.12) [62]. 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 (2.12) has connections to the (generalized) likelihood ratio test, where H_1 is the composite hypothesis $\boldsymbol{s} \in S$. 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 chapter, 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 (2.12)-(2.13) 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 (2.12)-(2.13) under Neyman-Pearson can be observed by noting that the decision statistic $\max_{\boldsymbol{s}\in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle$ is not a monotone function of the likelihood ratio, which in i.i.d. Gaussian noise for example, takes the form

$$\lambda(\boldsymbol{x}) = \int \exp\left(\frac{\langle \boldsymbol{s}, \boldsymbol{x} \rangle - \|\boldsymbol{s}\|^2/2}{\sigma^2}\right) \mathrm{d}\nu(\boldsymbol{s}). \tag{2.14}$$

FIG 2.3 and 2.4 illustrate such suboptimality for a particular problem configuration in $\mathbb{R}^2$. Note that throughout this and the following chapter, 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 the true positive rate TPR $\equiv 1 -$ FNR. This highlights the connection to the notion of error rates in machine learning, and also facilitates demonstration of the curves and axis ranges at very low error rates.

It is somewhat unsurprising that matched filtering is suboptimal in this setting, since the decision rules (2.12)-(2.13) do not make use of the prior ν, while the likelihood ratio test assumes (and uses) this prior. However, the matched filtering rule (2.12)-(2.13) is also in general suboptimal in the "prior-free" minimax setting. Consider the scenario in FIG 2.5 as an example, where the signal space $S \subset \mathbb{R}^2$ 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.

We also note that this suboptimality is, in some sense, not because we don't have sufficient templates. In the example shown in FIG 2.5, the matched filtering model already covers the entire signal set which consists of two signals. Furthermore, we will see in the later discussions that

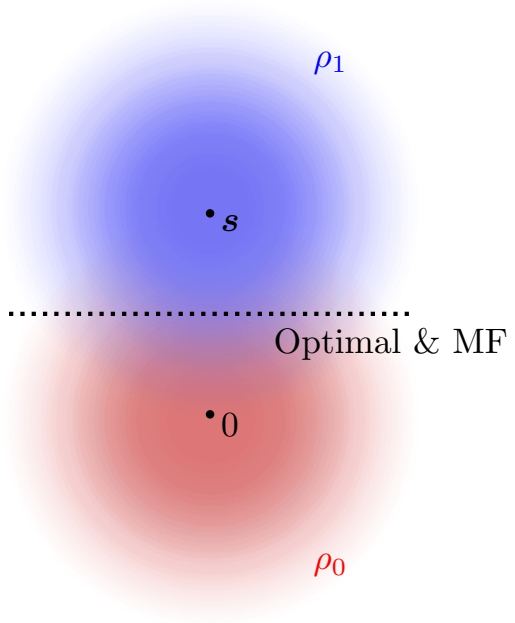

Figure 2.2: Optimality of matched filtering in single signal detection.

matched filtering has other structural limitations when working with non-Gaussian noise distributions.

2.4 From Matched Filtering to Neural Networks

Since the matched filtering rule (2.13) 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 (2.13) is equivalent to a feedforward neural network.

2.4.1 Neural Networks: Notation and Basics

A *neural network* implements a mapping from the signal space $\mathbb{R}^{D_{\text{in}}}$ to an output space $\mathbb{R}^{D_{\text{out}}}$:

$$f_{\boldsymbol{\theta}} : \mathbb{R}^{D_{\text{in}}} \to \mathbb{R}^{D_{\text{out}}}. \tag{2.15}$$

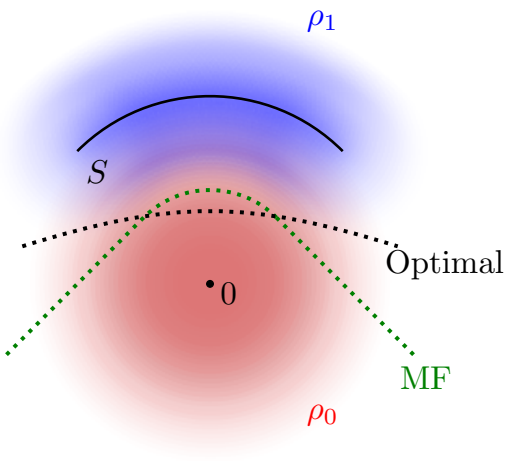

Figure 2.3: Suboptimality of matched filtering under the Neyman-Pearson framework.

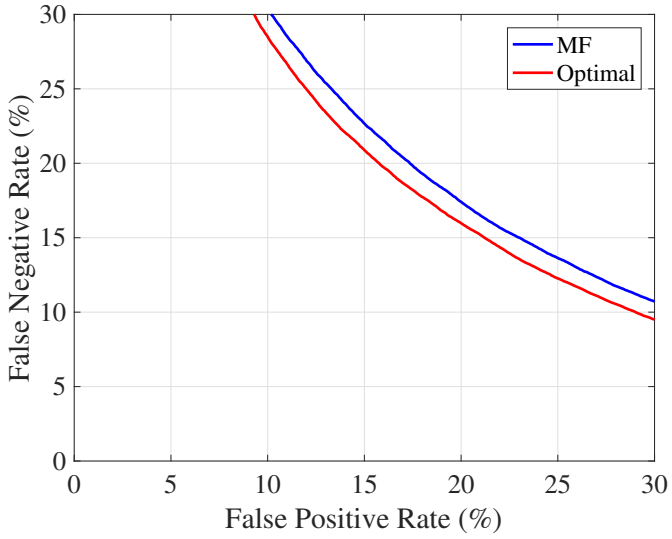

Figure 2.4: Comparison of ROC curves of the optimal classifier and matched filtering in the 2-dimensional concept as illustrated in FIG 2.3.

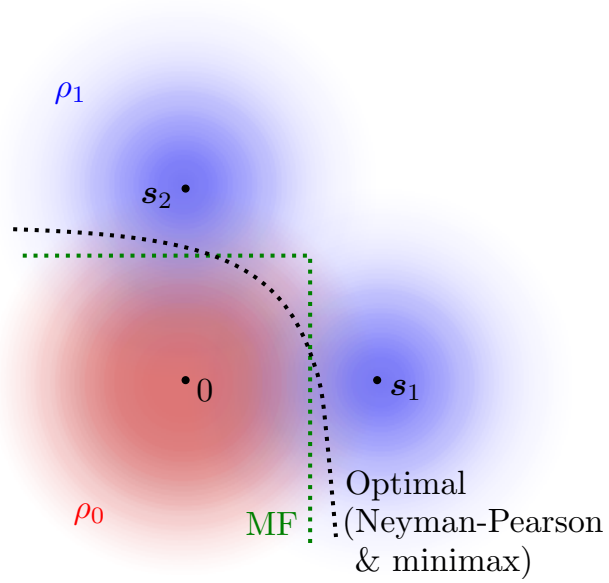

Figure 2.5: Suboptimality of matched filtering under the minimax framework.

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

$$\boldsymbol{x} \mapsto \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \tag{2.16}$$

followed by an element-wise activation function ϕ:

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{x}) \quad = \quad & \boldsymbol{W}^L \phi\Big(\boldsymbol{W}^{L-1}\phi\Big(\ldots \phi\Big(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1\Big) \\ & \qquad \ldots\Big) + \boldsymbol{b}^{L-1}\Big) + \boldsymbol{b}^L. \end{aligned} \tag{2.17}$$

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

$$\phi([v_1, \ldots, v_n]^T) = [\phi(v_1), \ldots, \phi(v_n)]^T. \tag{2.18}$$

The intermediate products

$$\boldsymbol{\alpha}^\ell(\boldsymbol{x}) = \phi\Big(\boldsymbol{W}^\ell \phi\Big(\ldots \phi\Big(\boldsymbol{W}^1\boldsymbol{x} + \boldsymbol{b}^1\Big) \cdots + \boldsymbol{b}^\ell\Big)\Big) \tag{2.19}$$

are sometimes referred to as *features* [136]. 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 [137]. For example, *maximum pooling* takes the maximum of adjacent features. In our notation, we can denote this operation by ρ^ℓ and write

$$\boldsymbol{\alpha}^\ell(\boldsymbol{x}) = \rho^\ell \phi\left(\boldsymbol{W}^\ell \boldsymbol{\alpha}^{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}^\ell\right), \tag{2.20}$$

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** $n^0, \ldots, n^L$*,* **pre-activation dimensions** $m^1, \ldots, m^L$*,* **parameters**

$$\begin{aligned} \boldsymbol{\theta} \;=\; \Big(& \boldsymbol{W}^L \in \mathbb{R}^{m^L \times n^{L-1}}, \ldots, \boldsymbol{W}^1 \in \mathbb{R}^{m^1 \times n^0}, \\ & \boldsymbol{b}^L \in \mathbb{R}^{m^L}, \ldots, \boldsymbol{b}^1 \in \mathbb{R}^{m^1} \Big), \end{aligned} \tag{2.21}$$

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*

$$[\rho^\ell]_i(\boldsymbol{v}) = \max_{j \in I_i^\ell} \boldsymbol{v}_j, \tag{2.22}$$

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

$$\boldsymbol{\alpha}^\ell(\boldsymbol{x}) = \rho^\ell \phi(\boldsymbol{W}^\ell \boldsymbol{\alpha}^{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}^\ell), \quad \ell = 1, \ldots, L. \tag{2.23}$$

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 [138]: universal approximation theorems indicate that *nonlinear* neural networks (with non-polynomial activation ϕ) can accurately approximate any continuous function, as long as the network is sufficiently deep and/or wide [139, 140, 141]. 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 corresponds to the dimension of the signal manifold S [12].

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 [23]. 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 [23] or from harmonic analysis considerations [142]. 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 [143], magnetic resonance image reconstruction [144] etc.).

In the following sections, we will follow this approach: we will give two ways of interpreting the matched filtering decision rule (2.13) 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.

2.4.2 Matched Filtering as a Shallow Neural Network

In the language of the previous section, it is not hard to express the decision statistic (2.13) of matched filtering as a specific fully connected neural network with one layer ($L = 1$). Writing

$$\rho^1(\boldsymbol{z}) = \max_i z_i, \tag{2.24}$$

$$\phi(t) = t, \tag{2.25}$$

$$\boldsymbol{W}^1 = \begin{bmatrix} \boldsymbol{s}_1^* \\ \boldsymbol{s}_2^* \\ \vdots \\ \boldsymbol{s}_K^* \end{bmatrix} \in \mathbb{R}^{K \times D}, \tag{2.26}$$

$$\boldsymbol{b}^1 = \boldsymbol{0}, \tag{2.27}$$

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. we have

$$\max_i \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle = \rho^1 \phi \left(\boldsymbol{W}^1 \boldsymbol{x} + \boldsymbol{b}^1 \right). \tag{2.28}$$

In words, the features produced by this neural network correspond to the correlations of the input with the templates $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K$. FIG 2.6 illustrates this (simple) architecture, which we label `MNet-Shallow`.

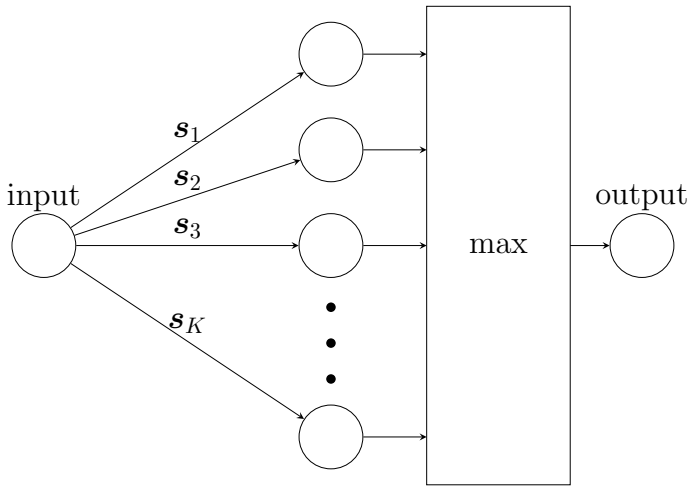

Figure 2.6: Illustration of `MNet-Shallow`. Bias terms are omitted in the illustration. (We note that for more complex networks arbitrary pooling operations can replace the "max" box.)

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

$$f_{\texttt{MNet-Shallow},\boldsymbol{\theta}}(\boldsymbol{x}), \tag{2.29}$$

where $\boldsymbol{\theta} = (\boldsymbol{W}^1, \boldsymbol{b}^1)$ represent the weights and biases. When these are chosen as in (2.26)-(2.27), `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 (2.13) with templates $\boldsymbol{s}_i$. 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 [140, 145].

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 2.7.

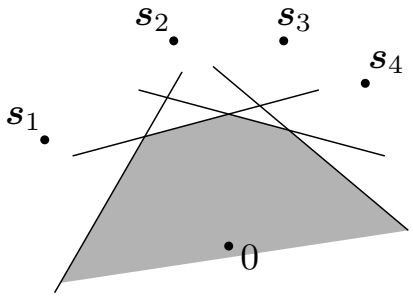

Figure 2.7: The set of points classified as noise by matched filtering and `MNet-Shallow` is always a convex set.

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 1. *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*

$$\{\boldsymbol{x} \mid \lambda(\boldsymbol{x}) \leq \tau\} \tag{2.30}$$

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

However, for general non-Gaussian distributions, the optimal decision region is often nonconvex. We illustrate this result in FIG 2.8. 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.

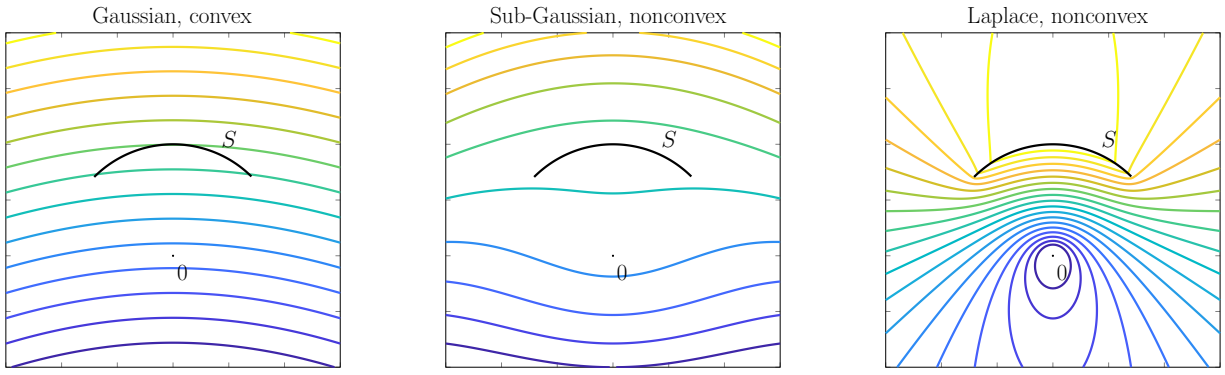

Figure 2.8: Contours of log likelihood ratio with various noise distributions, and whether the optimal decision regions with $\delta = 0$ is always convex. Yellow represents larger values and blue represents lower values. From left to right: (1) Gaussian distribution, convex; (2) Sub-Gaussian distribution $\rho_{\text{noise}}(\boldsymbol{x}) \propto \exp(-C\|\boldsymbol{x}\|^3)$, not necessarily convex; (3) Laplace distribution, not necessarily convex.

2.4.3 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 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 2.9.

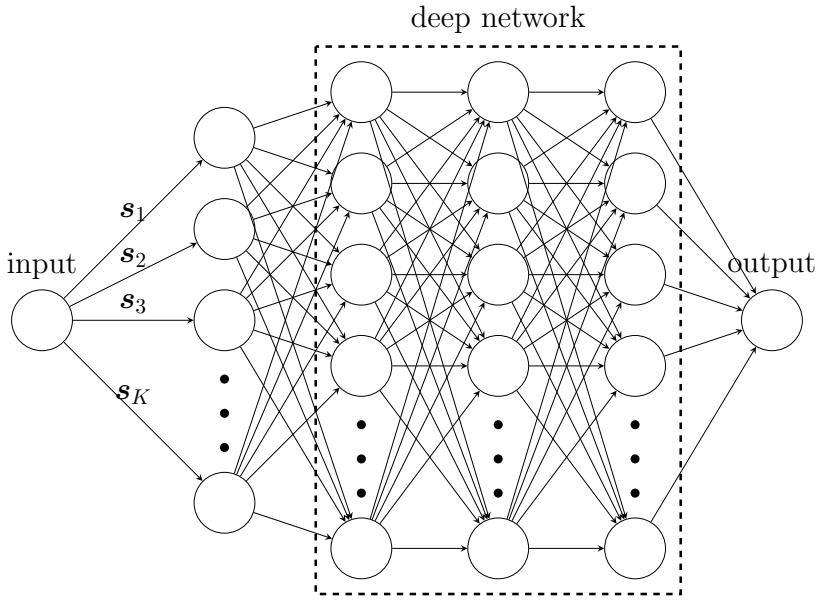

Figure 2.9: Illustration of `MNet-Deep`. Bias terms are omitted in the illustration. This network structure is obtained by replacing the max module in matched filtering (as in FIG 2.6) with a deep network.

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

$$\phi(t) = \max(t, 0). \tag{2.31}$$

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 $\langle \boldsymbol{s}_i, \boldsymbol{x} \rangle$. 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 3 ReLU units:

$$\max(a, b) = b + \phi(a - b) = \phi(b) - \phi(-b) + \phi(a - b). \tag{2.32}$$

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 $\boldsymbol{s}_i$, followed by this hierarchical structure for computing the maximum.

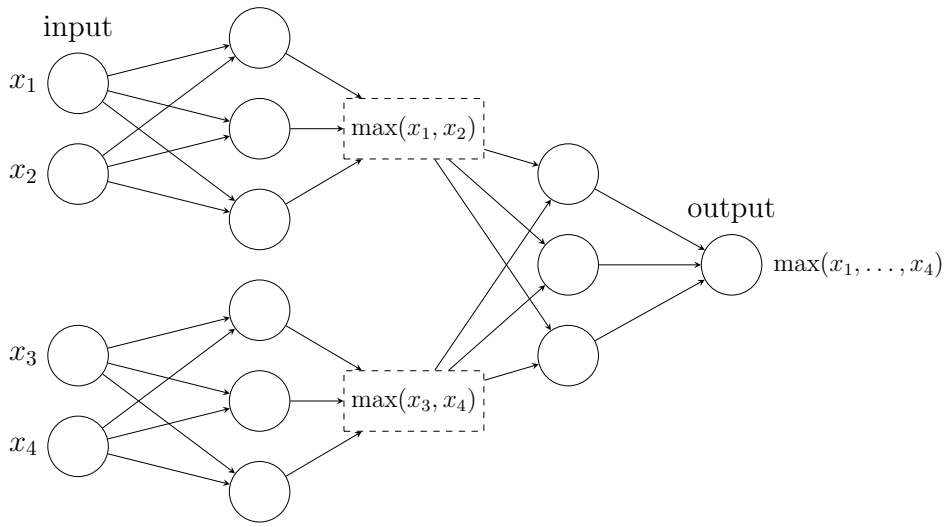

Figure 2.10: Illustration of implementing max with a ReLU network. The dashed boxes in the middle are not actual nodes in the network, but "imaginary" nodes to facilitate construction.

FIG 2.10 illustrates this hierarchical structure for the particular example of four inputs. The network in FIG 2.10 can be expressed as a ReLU network, with sparse weight matrices $\boldsymbol{W}^\ell$ ($\ell = 0, 1, 2$) for the layers respectively:

$$\boldsymbol{W}^0 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{W}^2 = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}, \tag{2.33}$$

$$\boldsymbol{W}^1 = \boldsymbol{W}^0 \otimes \boldsymbol{W}^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{2.34}$$

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

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 [140, 145].

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

$$f_{\texttt{MNet-Deep},\boldsymbol{\theta}}(\boldsymbol{x}), \tag{2.35}$$

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 non-ideal properties.

2.4.4 Equivalence of Matched Filtering and Neural Networks

We have demonstrated by construction the following claim:

Given any collection of templates $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K$ (for any $k \geq 1$), one can analytically determine weights $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_d$ such that

$$f_{\texttt{MNet-Shallow},\boldsymbol{\theta}_s}(\boldsymbol{x}) = \max_{i=1\ldots K} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle \tag{2.36}$$

$$f_{\texttt{MNet-Deep},\boldsymbol{\theta}_d}(\boldsymbol{x}) = \max_{i=1\ldots K} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle \tag{2.37}$$

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

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 [146, 147]. We will look for the most appropriate measure of complexity for this problem, and provide detailed analysis in future studies.

2.5 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 labelled samples

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N), \tag{2.38}$$

each of which consists of an observation $\boldsymbol{x}_i \in \mathbb{R}^D$ 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 $\boldsymbol{s}_i \sim \nu$, and setting $\boldsymbol{x}_i = \boldsymbol{s}_i + \boldsymbol{z}_i$, where $\boldsymbol{z}_i$ is observed LIGO noise strain. We solve the following optimization problem:

$$\min_{\boldsymbol{\theta}} \ \mathcal{R}_N(f_{\boldsymbol{\theta}}) := \frac{1}{N} \sum_{i=1}^{N} \ell\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\Big). \tag{2.39}$$

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

$$y \log(f_{\text{sigmoid}}(\hat{y})) + (1 - y) \log(1 - f_{\text{sigmoid}}(\hat{y})). \tag{2.40}$$

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

$$f_{\text{sigmoid}}(x) = \frac{1}{1 + \exp(-x)} \tag{2.41}$$

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 $\boldsymbol{x}_i = \boldsymbol{s}_i + \boldsymbol{z}_i$ when $y_i = 1$ and $\boldsymbol{x}_i = \boldsymbol{z}_i$ when $\boldsymbol{y}_i = 0$, with $\boldsymbol{s}_i \sim \nu$, and $\boldsymbol{z}_i \sim \rho_{\text{noise}}$. Let

$$\mathcal{R}_\infty(f) = \mathbb{E}_{(\boldsymbol{x},y)}\ell(f(\boldsymbol{x}), y). \tag{2.42}$$

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 2. *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 $\hat{y} = y$.* [1] *Then the unique optimal solution $f_\star$ to the (functional) optimization problem*

$$\min_f \ \mathcal{R}_\infty(f) \tag{2.43}$$

is a strictly increasing function of the likelihood ratio λ:

$$f_\star(\boldsymbol{x}) = g(\lambda(\boldsymbol{x})), \tag{2.44}$$

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

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 is required? What are effective approaches to minimizing the empirical risk $\mathcal{R}_N$? In the next section we investigate these questions experimentally.

[1] 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.

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 (2.8). We convert the constrained problem (2.8) to an equivalent unconstrained problem,

$$\min_{\delta} \max_{\boldsymbol{s}\in S} \text{FNR}(\boldsymbol{s}) + c \cdot \text{FPR}, \tag{2.45}$$

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 signal space S at points $\{\boldsymbol{s}_i\}_{i=1}^N$. Since FPR does not depend on $\boldsymbol{s}$, it can be moved inside the maximization. Therefore, the minimax optimization problem can be transformed into

$$\min_{\delta} \max_{i=1,\ldots,N} \text{FNR}(\boldsymbol{s}_i) + \text{FPR}. \tag{2.46}$$

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

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\Big[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \neq y_i\Big], \tag{2.47}$$

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

$$\frac{1}{N}\sum_{i=1}^{N} \ell\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\Big). \tag{2.48}$$

This expression is similar to (2.39), but the difference is that all positive data in the dataset here are associated with ground truth $\boldsymbol{s}_{i_*}$.

2.6 Simulations and Experiments

2.6.1 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 [134], 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 is downsampled from the original 4096Hz to 2048Hz for processing efficiency. The downsampled L1 strain data is 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 [66, 72, 71, 70, 69, 68, 67], with the following parameters. *Approximant:* IMRPhenomD. *Mass range:* 40 to 50 $M_{\odot}$, uniformly distributed. *Spin:* 0. *Sampling rate:* 2048Hz. *Low frequency cutoff:* 30Hz. *Coalescence phase:* 0. *Polarization:* plus [148]. 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 [82, 149, 54, 150, 151], 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 is used to generate training and test datasets of positive and negative labelled 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 0.3s, namely at the location of 0.42s within the entire segment of 0.6s. 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 30Hz and 400Hz, whitening using a power spectral density estimated from the L1 strain data, and finally truncating to keep only the center 0.3 second (614 samples).

2.6.2 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 2.11. 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.

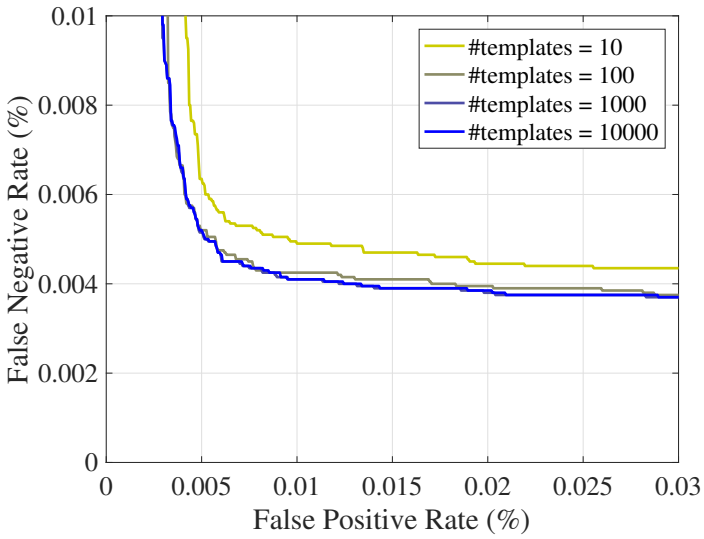

Figure 2.11: The best performance of matched filtering with given number of templates across 30 independent runs. The performance starts to saturate above 1000 templates.

2.6.3 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 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 [152], and a constant learning rate of 10^{-5}.

2.6.4 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. FIG 2.12 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 2.13 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 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.

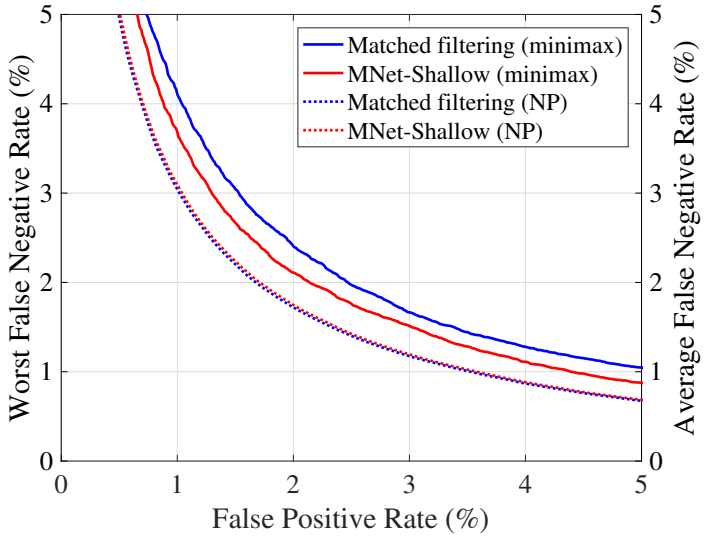

Figure 2.12: ROC curves of the trained shallow neural network and matched filtering. The solid curves correspond to the vertical axis on the left, and the dotted curves correspond to the vertical axis on the right. For both models we show both the worst (minimax) performance and the average performance 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.

2.7 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 pre-specified family of templates. This should be investigated in the future.

The proposed architectures can be adapted to time-varying noise distributions, by pre-training on very large collections of (synthetic) Gaussian noise and then adapting the pre-trained network using a smaller number of online examples. This kind of pre-training may also be helpful in

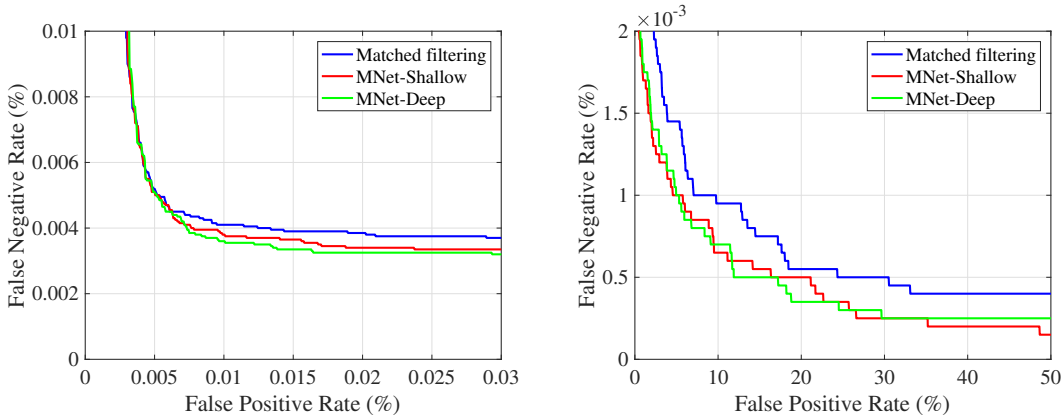

Figure 2.13: ROC curves of the trained `MNet-Shallow` and `MNet-Deep` models compared with matched filtering. Left and right panels plot the same curves, but have different axis ranges to better show the contrast between the curves.

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 (overparameterized) 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 don't 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 gravitational wave detectors worldwide as a single detector and analyzing data from multiple gravitational wave 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.

2.8 Conclusion

In this work, 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 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.

Chapter 3: Boosting the Detection Efficiency with Hierarchical Neural Networks

3.1 Introduction

In this chapter, we turn our focus to reducing the computational complexity of the detection methods, and show that by utilizing structures within the detection problem, we can construct trainable architectures with significantly higher computational efficiency.

As we discussed in the introductory chapter of this thesis, the computational complexity of a gravitational wave search method is closely related to its online processing capacity, and directly affects the possible scope of the search. In the literature, a promising approach to reducing the complexity of matched filtering searches has been to apply a two-step hierarchical search, which seeks to rapidly reject most negative samples [153]. Later, [154] expands the hierarchy to involve temporal multi-scale approach. Some other meritorious extensions include using geometric template placing [27] and hierarchy based on chirp times [155]. The work of [156] applies two-step detection within the PyCBC framework, and compares the performance on simulated data. A recent work of [157] further combines the two-step method with dimensionality reduction in the template space using principal component analysis (PCA). All the above examples demonstrate improvements relative to basic matched filtering in various settings for gravitational wave detection.

Similar ideas have also been widely explored and applied in machine learning contexts. For example, [158, 159] consider hierarchical matching of image features in both spatial domain and feature domain for image classification. [160] considers image classification using hierarchical matching in the spatial domain. In natural language processing, hierarchical model have also been used in sentiment classification [161]. More specific applications of this idea include medical

imaging [162], human detection and segmentation [163], and crime classification [164].

In the meantime, with the growing literature of applying deep learning and neural networks on gravitational wave detection, it is tempting to leverage deep learning's power to reduce complexity. Indeed, various neural network architectures have been shown to perform tasks such as gravitational wave detection, parameter estimation, noise transients identification and data denoising [88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 92, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123], at performance levels comparable to that of matched filtering. Furthermore, it has been shown that matched filtering is generally suboptimal for parametric signal detection [165, 125], and the performance can be improved by optimizing the templates using deep learning techniques [125]. This can be achieved by setting up a neural network that is formally equivalent to matched filtering, and then training on data. Inspired by the flexibility of deep learning models, it is conceptually appealing to explicitly incorporate computational efficiency into the neural network objectives, aim to achieve "the best of both worlds."

Recall the MNet architectures proposed in the previous chapter, where the weights of the network $\boldsymbol{s}_i$ are initialized as templates and then trained over data, creating an advantage over classic matched filtering. If we compare its computational efficiency against matched filtering (measured in terms of the number of operations required to achieve a target error rate), the strict performance improvement with identical architecture suggests that one can expect a strict efficiency improvement as well. However, the structural similarity between MNet-Shallow and matched filtering implies that such efficiency gains may typically be very limited. In order to achieve efficiency gains on higher orders of magnitude, we may need to reconsider the parametric detection problem, and innovate on the basic matched filtering rule. As we present in the next section, one solution is to arrange the templates in a multi-layer hierarchy, so that significant proportions of negative-labeled data are subject to early rejections.

In this work, we propose a novel neural network architecture, named Hierarchical Detection Network (HDN), which takes the form of a multi-layer matched filtering with trainable parameters.

In order to achieve the dual goal of accuracy and efficiency, we constructed a *novel loss function* that explicitly incorporates computational complexity. We demonstrate the efficiency gains on data with open LIGO noise data and synthetic gravitational wave signal injections. As a quick glance at the performance gains, when tested on data with synthetically injected signals at signal-to-noise ratio (SNR) 9, compared with matched filtering, two-layer HDN can achieve false positive and false negative rates 0.2% with 79% lower complexity, and reduces error rates by 88% when at equal complexity equivalent to 100 templates, for instance. Experimental details are described in Section 3.4.

Yet, the two-layer networks do not reveal the full power of the proposed model. We further show that by training a three-layer model with careful initialization, it is capable of achieving even better accuracy at lower complexity. We also provide some intuitive insights into the mechanism behind multi-layer hierarchical models and their construction.

The rest of the chapter is organized as follows. Section 3.2 introduces Hierarchical Detection Networks, including the setup, complexity and training process. Section 3.3 further discusses the complexity reduction from HDN. Section 3.4 presents experimental results of applying HDN on real LIGO data and synthetic injections. We discuss some further implications and future steps of this work in Section 3.5.

3.2 Hierarchical Detection Networks

In this section, we present the Hierarchical Detection Network (HDN), which improves over matched filtering and MNet-Shallow to simultaneously maximize statistical performance and computational efficiency.

The main idea behind HDN is intuitive. If an input segment clearly contains no gravitational wave signals, we may not need to subject it to millions of templates to tell that. A small number of "gatekeeper" templates may be sufficient for confidently rejecting these "obviously wrong" instances. Once these inputs have been ruled out, we can apply a more refined test using possibly more templates, and reject a larger portion of the input space. This procedure can be repeated,

until in the very last step, we employ our full template bank for a full diagnosis on the remaining instances which all previous tests failed to reject. Since the overwhelming majority part of the gravitational wave strain data contains noise only, most instances will likely be addressed by the initial simple layers of the model, saving the need for the full template bank. In addition, different layers of the HDN may be designed to specialize in different parts of the input space, such that the available parameter space of the potentially allowed waveforms are successively restricted as the hierarchical process progresses from later to layer, allowing for further efficiency gains.

3.2.1 Architecture of HDN

We first formally define a hierarchical detection network (HDN). Generally speaking, a HDN is a hierarchical template matching model trained as a neural network, as illustrated in Fig 3.1. Let L be the number of layers in the hierarchical structure, and let $\{\boldsymbol{s}_i\}_{i=1}^{K}$ be the set of K templates used by the model. For each layer $\ell = 1, \ldots, L$, only the first n_ℓ of these templates are used in that layer, where $0 < n_1 < \cdots < n_L = K$. Let the threshold associated with template i at layer ℓ be $t_{i,\ell}$, $i = 1, \ldots, n_\ell$. Here we let layer l reuse all templates from the previous layer(s), but assign independent threshold values to the reused templates at different layers, in order to reduce computation complexity.

Following conventions of the machine learning literature on binary classification, we call an input $\boldsymbol{x}$ *positive* if it contains a gravitational wave signal, and *negative* if it only contains noise. For a given input $\boldsymbol{x}$, the model processes it using the following procedure:

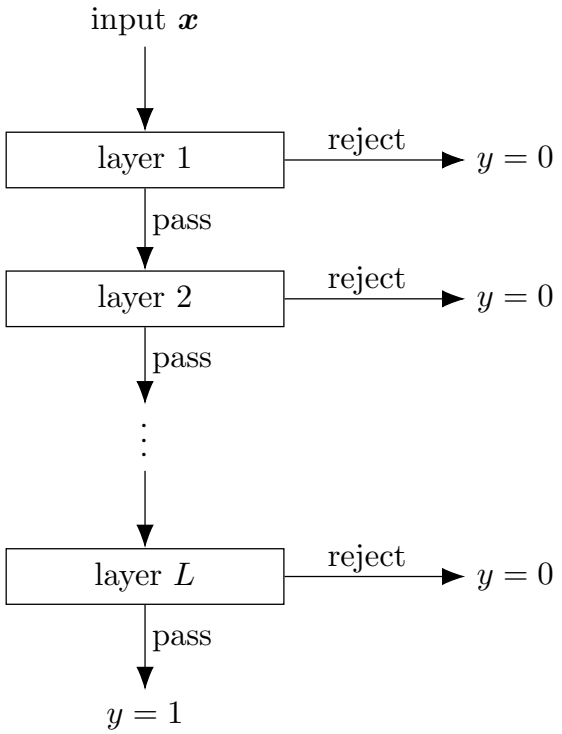

Figure 3.1: Illustration of a hierarchical detection network.

Algorithm 1 The HDN algorithm

Parameters: $L, K, \{n_\ell\}, \{\boldsymbol{s}_i\}, \{t_{i,\ell}\}$

Input: $\boldsymbol{x}$

$\ell \leftarrow 1$

while $\ell \leq L$ **do**

$y_\ell \leftarrow \max_{i \leq n_\ell} \langle \boldsymbol{x}, \boldsymbol{s}_i \rangle - t_{i,\ell}$

if $y_\ell < 0$ **then**

return 0

end if

$\ell \leftarrow \ell + 1$

end while

return 1

More formally, for a given input $\boldsymbol{x}$, let

$$y_\ell = \max_{i \le n_\ell} \langle \boldsymbol{x}, \boldsymbol{s}_i \rangle - t_{i,\ell} \tag{3.1}$$

be the matching output at layer ℓ, $\ell = 1, \ldots, L$. Let

$$I_\ell = \begin{cases} \mathbb{1}[y_\ell > 0], & \text{if } I_{\ell-1} = 1 \\ 0, & \text{if } I_{\ell-1} = 0 \end{cases} \tag{3.2}$$

be the indicator of whether the input passes layer ℓ of the model on to the later layer(s), $\ell = 1, \ldots, L-1$, and define $I_L \equiv 0$. With these notations, the overall output of the model can be written as

$$y(\boldsymbol{x}) = \sum_{\ell=1}^{L} y_\ell (1 - I_\ell) \prod_{k=1}^{\ell-1} I_k. \tag{3.3}$$

Note that both matched filtering and MNet-Shallow can be unified under the framework of HDN, viewed as a model with a single layer. In the meantime, some of the existing two-step MF methods [153, 27, 156] can also be interpreted under this framework. Furthermore, the HDN architecture is not restricted to the typical two-layer hierarchy of coarse and fine searches, but can utilize multiple layers which specialize in different parts of the signal space. The use of multiple layers and their setup is further discussed in Section 3.3.

3.2.2 Measure of Computational Complexity

With matched filtering and HDN unified under the same framework, we can provide a formal definition of computational complexity to facilitate our discussion. We are often most concerned about the execution efficiency of the model in deployment rather than in training, since it determines the real-time processing abilities. In the meantime, any computational cost of setting up the parameters of the model, including template selection for matched filtering and training for neural networks, is a one-time cost and can be conducted offline. Therefore it is natural to define

complexity based on test time.

Also, since for the vast majority of time the input strain does not contain gravitational wave events, we can capture the computational complexity solely by its performance on negative data. This leads to the following definition of complexity:

Definition 2 (Complexity)**.** *The (computational) complexity of a HDN model is defined as the expected number of template matching (inner product) operations conducted to evaluate a negative input.*

Formally, we can write the complexity as

$$Z = \mathbb{E}_{\boldsymbol{x}\sim F_-}[z(\boldsymbol{x})], \tag{3.4}$$

where

$$z(\boldsymbol{x}) = \sum_{\ell=1}^{L} n_\ell (1 - I_\ell) \prod_{k=1}^{\ell-1} I_k \tag{3.5}$$

is the number of matching operations required for evaluating an input $\boldsymbol{x}$.

To illustrate this measure of complexity, note that for matched filtering and MNet-Shallow models, the complexity simply equals the number of templates used in the model. For a two-layer HDN, assuming only a proportion p of negative data enters the second layer, the complexity for the model will be $n_1 + p \cdot n_2$. Intuitively, if the initial layer contains fewer templates while being able to reject a significant portion of negative inputs, these inputs will not need to undergo the entire model, hence reducing the complexity of the model. This straightforward idea forms the basis of HDN, upon which we further leverage the power of data through training for an additional boost in performance.

3.2.3 Training of HDN

So far, we have described the behavior of HDN at test/deployment time, and now we turn our attention to the training process. Conceptually, we want to set up a loss function as an appropriate combination of classification error and model complexity, so that minimizing the loss would

achieve simultaneously accuracy and efficiency. However, a loss function directly based on the above expressions (3.3) and (3.5) is undesirable because of non-differentiability. Instead, we use soft surrogates for the indicators I_ℓ. Define

$$\hat{I}_\ell = \phi(y_\ell) \tag{3.6}$$

for $\ell = 1, \ldots, L-1$ where $\phi(x) := \frac{1}{1+e^{-x}}$ is the sigmoid function, serving as a soft surrogate of the step function. Also let $\hat{I}_L \equiv 0$. Note that during training we can simply compute y_ℓ for all layers regardless of whether previous layers were passed, since this will only be a one-time offline cost. Define the soft surrogates for $y(\boldsymbol{x})$ and $z(\boldsymbol{x})$ accordingly:

$$\hat{y}(\boldsymbol{x}) = \sum_{\ell=1}^{L} y_\ell (1 - \hat{I}_\ell) \prod_{k=1}^{\ell-1} \hat{I}_k, \tag{3.7}$$

$$\hat{z}(\boldsymbol{x}) = \sum_{\ell=1}^{L} n_\ell (1 - \hat{I}_\ell) \prod_{k=1}^{\ell-1} \hat{I}_k. \tag{3.8}$$

Assume the training dataset is $\{(\boldsymbol{x}_i, y_i^\star)\}_{i=1}^N$, with N_+ positive entries and N_- negative entries. The loss function can be formulated as

$$\mathcal{L} = \frac{1}{N} \sum_i \ell_i^{\text{accu}} + \lambda \cdot \frac{1}{N_-} \sum_{i: y_i^\star = 0} \ell_i^{\text{cplx}}, \tag{3.9}$$

where

$$\ell_i^{\text{accu}} = y_i^\star \log p_i + (1 - y_i^\star) \log(1 - p_i) \tag{3.10}$$

with $p_i = \frac{1}{1+e^{-\hat{y}_i}}$ is equivalent to the cross-entropy loss for binary classification, and

$$\ell_i^{\text{cplx}} = \hat{z}_i \tag{3.11}$$

is the soft approximate for the complexity of evaluate the negative inputs. With the loss function (3.9) defined above, we can then train the model parameters $\{\boldsymbol{s}_i\}$ and $\{t_{i,\ell}\}$ using first order

optimization methods.

Experimental results of the HDN architecture will be shown in Section 3.4.

3.3 Complexity Reduction from Multiple Layers

Here we provide some heuristic insights into why the hierarchical model achieves reduced complexity at similar target performance levels, particularly with more layers.

Consider as an example a two-layer hierarchical model with n_1, n_2 templates respectively on the layers. Let α_ℓ and β_ℓ denote respectively the false positive rate (FPR) and false negative rate (FNR) of layer ℓ conditioned on data that reaches the corresponding layer. Recall that FPR denotes the proportion of negative samples that are falsely classified as positive, and FNR the proportion of positive samples falsely classified as negative. The overall FPR, FNR and the complexity z can then be represented as:

$$\alpha_{\text{all}} = \alpha_1 \alpha_2 \tag{3.12}$$

$$\beta_{\text{all}} = \beta_1 + (1 - \beta_1)\beta_2 \tag{3.13}$$

$$z_{\text{all}} = n_1 + \alpha_1 (n_2 - n_1) \tag{3.14}$$

To understand why an improvement in complexity can be expected, we consider the following example setup of parametric detection as shown in FIG. 3.2. The probability density of the two labeled classes are ρ_0 and ρ_1 respectively. Note that the density for the negative class ρ_0 is precisely the noise density defined previously, and ρ_1 is the convolution of ρ_0 and the signal density. Imagine a baseline MF model with decision boundary as shown by the green curve, at the cost of n templates, where n has to be relatively large to approximate the smoothly curved boundary. Then we can construct the following hierarchical model to achieve a significantly lower complexity with identical statistical accuracy. To do this, we construct a simple two-layer hierarchical model, with the first layer decision boundary as shown by the dotted blue line, and the second layer decision boundary coinciding with that of the MF model. Notice that the first layer features a very low

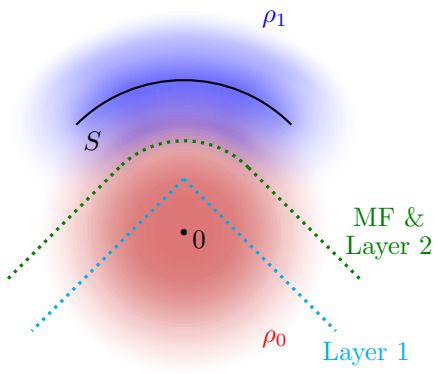

Figure 3.2: An example of the complexity advantage of hierarchical detection models.

complexity n_1 (with $n_1 = 2$ in this example), and in the meantime has a fairly high true negative rate $1 - \alpha$. Since the second layer reproduces the MF decision boundary, the overall decision rule of the hierarchical model is identical to that of the MF model, and hence they share exactly the same ROC (receiver operating characteristic) curves. However, the complexity of the HDN model is $n_1 + \alpha_1(n_2 - n_1)$, which is significantly smaller than n_2 provided n_1 is small compared with n_2 and α_1 is not too close to 1.

This example provides inspirations for a general recipe for designing hierarchical models with reduced complexity. For any decision rule given by a MF model, we can construct a sequence of preceding layers whose negative decision regions all lie inside the negative decision region of the MF, and finally let the very last layer be equivalent to the original MF. The resulting hierarchical model will again have exactly the same overall decision rule and hence ROC curve, but with significantly reduced complexity. An illustrating example is shown in FIG. 3.3.

More generally, such constructions of hierarchical models can serve as good initializations for a HDN. One practical initialization scheme for an L-layer HDNs is the following: first train a separate $L - 1$-layer model that only consists of the latter $L - 1$ layers of the desired model. Then we initialize the latter $L - 1$ layers of the original model with the trained network, and initialize the first layer with small $t_{i,1}$ values such that almost all inputs pass. This gives an initialization of the L-layer model which at initialization essentially replicates the $L - 1$-layer model. From there, we train the initialized L-layer model on data, which will leverage the higher architectural capacity

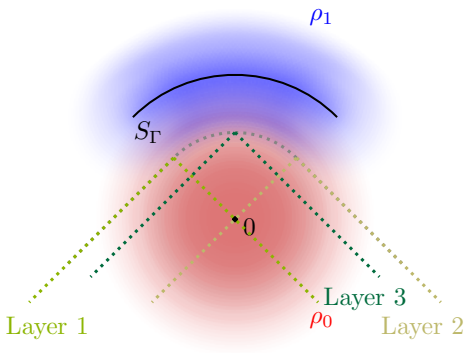

Figure 3.3: A hierarchical model with more simple layers that lie inside the overall negative decision region.

for further improved performance and complexity. An experiment that illustrates this approach is shown in Section 3.4.

3.4 Simulation and Experiments

3.4.1 Data Generation

In the experiments, we use open L1 strain data from LIGO Livingston's O2 run between August 1 and August 25, 2017 with ANALYSIS_READY flag [166]. The total duration of the frame files is 389.12 hours. We downsample the strain data from the original 4096Hz to 2048Hz for processing efficiency. The downsampled data is then divided into segments of 2 seconds, with each segment overlapping with 50% of its preceding segment.

To evaluate the accuracy of detection models, we need both positive and negative datasets. For the negative datasets, the strain data itself is used. For the positive datasets, due to the very limited number of confirmed detections of gravitational wave events, we generate positive data by injecting synthetic waveforms into the noise strains, at a preset SNR value.

The entire L1 strain dataset is first divided into two sets to be used in training and test respectively, such that any segment in the training set does not overlap with any segment in the test set. For training and test respectively, a positive and a negative dataset are generated. For the positive datasets, synthetic waveforms are generated with masses m_1, m_2 uniformly drawn from $[20, 50]$

(times solar mass $M_\odot$) and 3-dimensional spins drawn from an isotropic distribution and with spin dimensionless magnitudes drawn from a uniform distribution within [0,1]. One waveform is injected to each 2-second segment of data, and the injected waveforms are aligned such that the peak is located at 0.95 second, and the injection amplitude is chosen such that the signal-to-noise ratio (SNR) after preprocessing is constant at 9. The preprocessing is applied to all data (after injection if applicable) by using a finite-impulse-response (FIR) bandpass filter with cutoff frequency 30Hz and 400Hz, whitening with power spectral density estimated from the L1 strain data, and truncating to only keep the center 1 second.

3.4.2 Two-Layer Networks

In this experiment, we limit our HDN models to two layers and $n_2 = 10n_1$. At initialization, the templates $\boldsymbol{s}_i$ are chosen as random gravitational waveforms from the same parameter space, and the thresholds $t_{i,\ell}$ are set to the same within each of the two layers. The parameter λ is the loss function is fixed at $\lambda = 10^{-4}$. For the optimization procedure of the network, we use the Adam optimizer [152] which is common in modern deep learning, and a constant learning rate (i.e. scaling of the update at each iteration) of 10^{-4}.

FIG. 3.4 shows the comparison of the complexity-performance trade-offs of MF and HDN models, where the HDN models are two-layer architectures structured as described above. The horizontal axis plots the logarithm of the complexity measure defined in this chapter, and the vertical axis plots the logarithm of error rates at the point on the ROC curve where FPR = FNR. This choice of measure eliminates the arbitrariness of choosing FNR at a fixed FPR level. For each architecture, 10 independent runs are conducted, and the one with lowest accuracy measure is shown. The blue curve for HDN is cut off early due to memory limitations of training the model. FIG. 3.5 visualizes the proportion of error rate reduced by HDN as compared with MF at equal complexities. We see that HDN consistently achieves a lower complexity than MF at equal accuracy, with higher advantages at lower complexities.

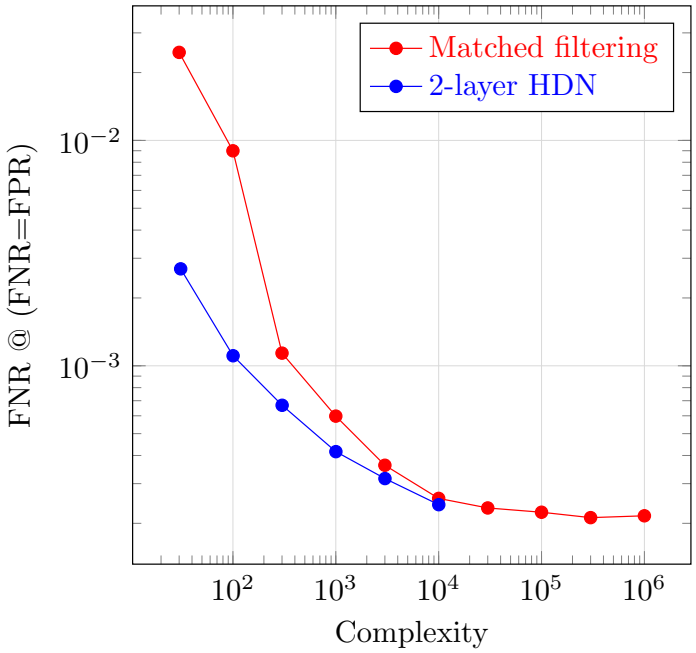

Figure 3.4: Complexity-performance trade-off of matched filtering and the hierarchical neural network.

3.4.3 Three-Layer Networks

We further demonstrate the power of the proposed model with a deeper three-layer network. Conceptually, since adding more layers strictly improves model expressability, it should never hurt performance provided that the parameters are initialized or trained appropriately.

In this experiment, we construct a 3-layer HDN with layer sizes $(n_1, n_2, n_3) = (30, 100, 1000)$ in the following way. First, a shallower 2-layer model with layer sizes $(100, 1000)$ is trained, and we use these trained parameters to initialize the latter two layers of the 3-layer model. We then initialize the first layer of the 3-layer model, setting the per-template thresholds $t_{i,1}$ as the same value for all i, such that all training data passes this first layer at initialization. This scheme ensures that at initialization, the 3-layer model essentially replicates the performance of the trained 2-layer model, giving it a head start before entering the training phase. The training is done in the same way as described before. FIG. 3.6 illustrates the architecture of this 3-layer model, along with the histograms of layer outputs of test data that reaches that layer, where the x-axis denotes the layer

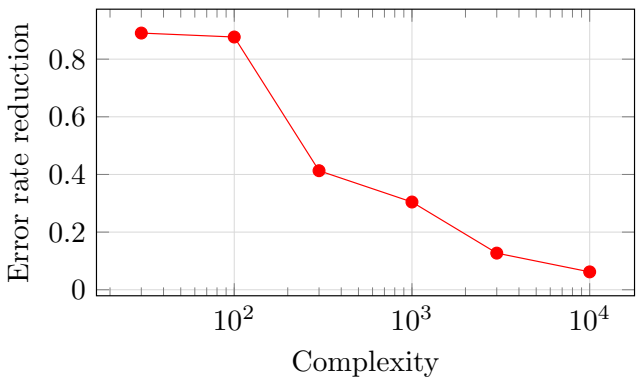

Figure 3.5: Proportion of error rate reduced by using HDN over MF.

output and the y-axis denotes the frequency of occurrence. The histograms are divided into positive and negative according to the true class labels. Specifically, the densities of layer 1 involves all input data, and the densities of layers 2 and 3 involve only the data that pass the previous layers. We see that most of the negative data are successfully intercepted by the initial layers, with very few of them reaches the final layer, which corroborates our intuition.

Here when evaluating the ROC curve, we adopt a slightly different approach that is more consistent with deep hierarchical models. Notice from equation (3.2) that the model uses a built-in threshold 0 to control the passing of each layer. When generating the ROC curve using a varying threshold, such a threshold should be applied at all layers instead of only the last layer. Therefore, at test time only, we replace the threshold 0 in equation (3.2) with a variable threshold $t \in \mathbb{R}$ which is constant for all layers, and compute the test outputs using (3.3) as before for each t value. Varying this threshold t produces the ROC curve. Also note that t determines which test entries would pass the layers, hence it also affects the model complexity evaluated on the negative test dataset. In actual deployment, the threshold t should be fixed at some level that gives the desired trade-off between FPR and FNR, so this is only for demonstration purpose.

FIG. 3.7 shows the comparison of ROC curves between a matched filtering model with 100 templates, a 2-layer HDN model with complexity 100.5 (used for initializing the 3-layer model), and a 3-layer HDN model with complexity 37.1 at the point of equal FPR and FNR. While the complexity of the 3-layer model depends on the specific point chosen on the ROC curve, it does

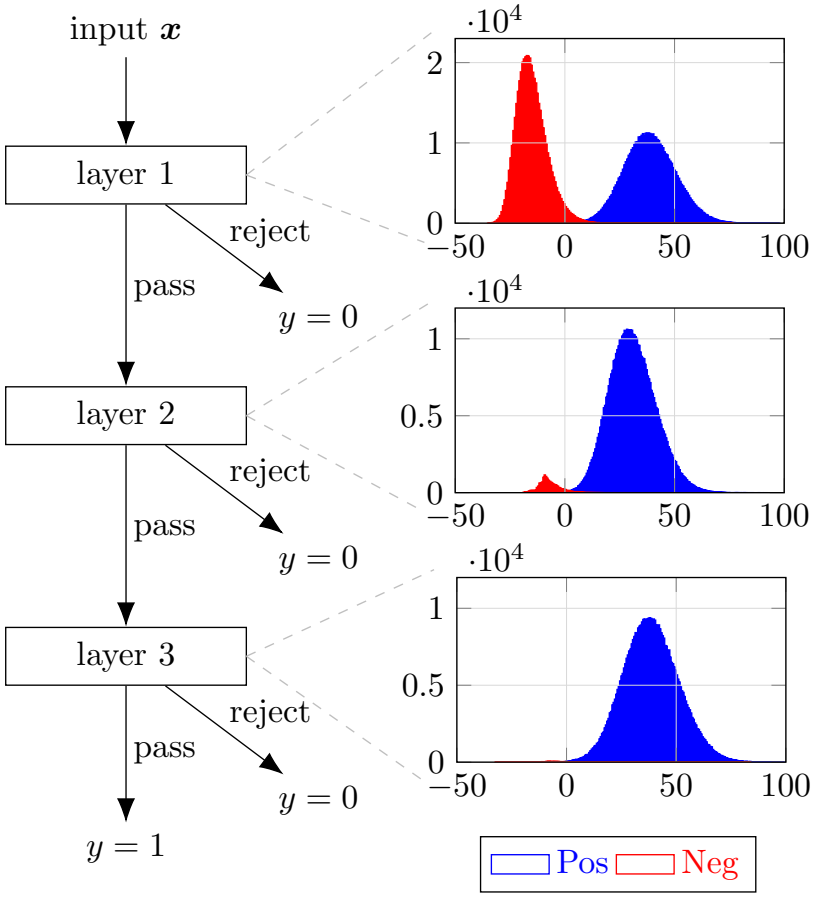

Figure 3.6: Illustration of the 3-layer architecture, and the output densities on the test data from each layer. Only data entries that reach a given layer is shown. We see that each layer successfully rejects the vast majority of incoming negative data, and barely any negative data reaches the last layer.

not exceed 65 for the entire segment of ROC curve shown in the figure, and is thus always lower than the 2-layer model. We see that the deeper 3-layer model excels at both accuracy and efficiency compared with the 2-layer model, and significantly more so if compared with the matched filtering model. This further showcases the power of depth in hierarchical models, and corroborates our discussion in Section 3.3.

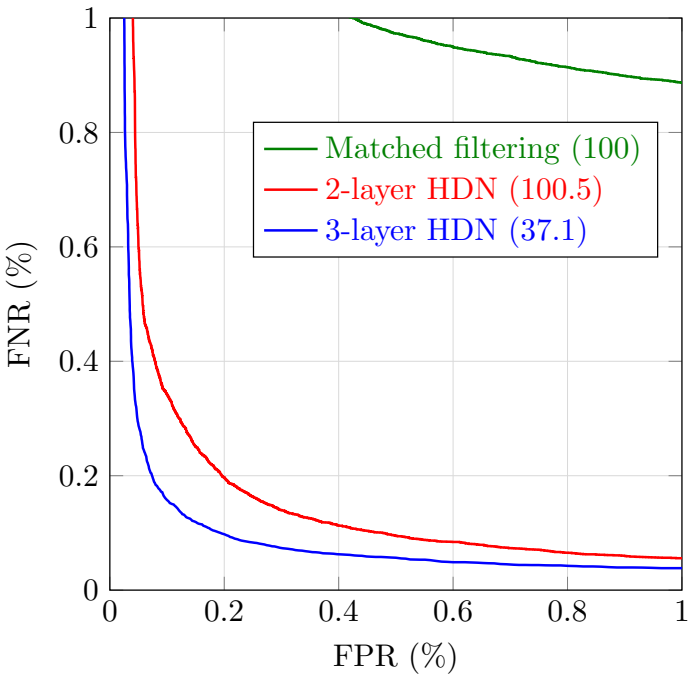

Figure 3.7: Comparison of ROC curves between three models. The numbers in parentheses show the complexity of the model.

3.5 Discussion

In this work, we showed that by leveraging ideas from classical matched filtering and modern machine learning, we are able to design systems for gravitational wave detection that simultaneously optimize statistical accuracy and efficiency. This general conceptual idea of trainable hierarchical matched filtering can be applied upon a wide range of existing proposals for efficient detection pipelines.

While the proposed HDN model conducts hierarchical rejection on the data, an alternative can be proposed to conduct hierarchical acceptance, namely to progressively label parts of the data as positive rather than negative. This has the advantage of aligning better with the matched filtering routine, since it suffices to use one matching template to confirm a signal. In the specific problem of gravitational wave signal detection, due to the class imbalance from the scarcity of actual gravitational wave events, the majority of computational complexity hinges on the classification

of negative data, and therefore a hierarchical rejection model will have much more significant efficiency gains. In more general signal detection problems, hierarchical acceptance constructions can also be deployed in similar fashions as HDN.

The proposed HDN can potentially have wider applications within the field of gravitational wave science. For example, in the task of glitch detection and identification [110, 167, 168, 169, 170, 171], one can combine existing constructions of machine learning based models with hierarchical models, to improve on both efficiency and accuracy.

One aspect to be further explored is how to select the number of layers in the hierarchy. While having more layers can boost model expressability and further leverage the efficiency gains, excessive hierarchy may offer diminishing returns, and also make training increasingly difficult. In the work we demonstrated how a 3-layer model excels over a 2-layer one, and there may be a "sweet spot" number of layers for a given signal detection setups. Another promising direction would be to incorporate prior knowledge about the signal domain such as low-dimensionality and representative features into the detection model, which may be able to further outperform these current models agnostic of the signal space properties.

Chapter 4: TpopT: Efficient Trainable Template Optimization on Low-Dimensional Manifolds

4.1 Introduction

Low-dimensional structure is ubiquitous in data arising from physical systems: these systems often involve relatively few intrinsic degrees of freedom, leading to low-rank [172, 173], sparse [174], or manifold structure [175, 176, 177]. In this work, we study the fundamental problem of detecting and estimating signals which belong to a low-dimensional manifold, from noisy observations [178, 179, 180].

Perhaps the most classical and intuitive approach to detecting families of signals is *matched filtering* (MF), which constructs a bank of templates, and compares them individually with the observation. Due to its simplicity and interpretability, MF remains the core method of choice in the gravitational wave detection of the scientific collaborations LIGO [46, 47], Virgo [45] and KARGA [181], where massive template banks are constructed to search for traces of gravitational waves produced by pairs of merging black holes in space [27, 182, 183, 126]. The conceptual idea of large template banks for detection is also widely present in other scenarios such as neuroscience [28], geophysics [29, 30], image pose recognition [31], radar signal processing [32, 33], and aerospace engineering [34]. In the meantime, many modern learning architectures employ similar ideas of matching inputs with template banks, such as transformation-invariant neural networks which create a large number of templates by applying transformations to a smaller family of filters [35, 36, 37].

One major limitation of this approach is its unfavorable scaling with respect to the signal manifold dimension. For gravitational wave detection, this leads to massive template banks in deployment, and presents a fundamental barrier to searching broader and higher dimensional signal

manifolds. For transformation-invariant neural networks, the dimension scaling limits their applications to relatively low-dimensional transformation groups such as rotations.

This work is motivated by a simple observation: instead of using sample templates to cover the search space, we can search for a best-matching template via optimization over the search space with higher efficiency. In other words, while MF searches for the best-matching template by enumeration, a first-order optimization method can leverage the geometric properties of the signal set, and avoid the majority of unnecessary templates. We refer to this approach as template optimization (TpopT).

In many practical scenarios, we lack an analytical characterization of the signal manifold. We propose a nonparametric extension of TpopT, based on signal embedding and kernel interpolation, which retains the test-time efficiency of TpopT.[1] The components of this method can be trained on sample data, reducing the need for parameter tuning and improving the performance in Gaussian noise. Our training approach draws inspiration from unrolled optimization [189], which treats the iterations of an optimization method as layers of a neural network. This approach has been widely used for estimating low-dimensional (sparse) signals [190, 191] with promising results on a range of applications [192, 193, 194, 195]. The main contributions of this work are as follows:

- Propose trainable TpopT as an efficient approach to detecting and estimating signals from low-dimensional families, with nonparametric extensions when an analytical data model is unavailable.
- Prove that Riemannian gradient descent for TpopT is exponentially more efficient than MF.
- Demonstrate significantly improved complexity-accuracy trade-offs for gravitational wave detection, where MF is currently a method of choice.

[1] In contrast to conventional manifold learning, where the goal is to learn a representation of the data manifold [184, 185, 186, 187, 188], our goal is to learn an *optimization algorithm* on the signal manifold.

4.2 Problem Formulation and Methods

In this section, we describe the problem of detecting and recovering signals from a low-dimensional family, and provide a high-level overview of two approaches — matched filtering and template optimization (TpopT). The problem setup is simple: assume the signals of interest form a d-dimensional manifold $S \subset \mathbb{R}^D$, where $d \ll D$, and that they are normalized such that $S \subset \mathbb{S}^{D-1}$. For a given observation $\boldsymbol{x} \in \mathbb{R}^D$, we want to determine whether $\boldsymbol{x}$ consists of a noisy copy of some signal of interest, and recover the signal if it exists. More formally, we model the observation and label as:

$$\boldsymbol{x} = \begin{cases} a\,\boldsymbol{s}_\natural + \boldsymbol{z} & \text{if } y = 1 \\ \boldsymbol{z}, & \text{if } y = 0 \end{cases}. \tag{4.1}$$

where $a \in \mathbb{R}_+$ is the signal amplitude, $\boldsymbol{s}_\natural \in S$ is the ground truth signal, and $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Our goal is to solve this detection and estimation problem with simultaneously high statistical accuracy and computational efficiency.

Matched Filtering. A natural decision statistic for this detection problem is $\max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle$, i.e.

$$\hat{y}(\boldsymbol{x}) = 1 \iff \max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle \geq \tau \tag{4.2}$$

where τ is some threshold, and the recovered signal can be obtained as $\arg\max_{\boldsymbol{s} \in S} \langle \boldsymbol{s}, \boldsymbol{x} \rangle$. [2] *Matched filtering*, or template matching, approximates the above decision statistic with the maximum over a finite bank of templates $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n_{\text{templates}}}$:

$$\hat{y}_{\text{MF}}(\boldsymbol{x}) = 1 \iff \max_{i=1,\ldots,n_{\text{templates}}} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle \geq \tau. \tag{4.3}$$

The template $\boldsymbol{s}_i$ contributing to the highest correlation is thus the recovered signal. This matched filtering method is a fundamental technique in signal detection (simultaneously obtaining the esti-

[2] This statistic is optimal for detecting a single signal $\boldsymbol{s}$ in iid Gaussian noise; this is the classical motivation for matched filtering [196]. For detecting a family of signals $\boldsymbol{s} \in S$, it is no longer statistically optimal [197]. However, it remains appealing due to its simplicity.

mated signals), playing an especially significant role in scientific applications [27, 30, 28].

If the template bank densely covers S, (4.3) will accurately approximate (4.2). However, dense covering is inefficient — the number n of templates required to cover S up to some target radius r grows as $n \propto 1/r^d$, making this approach impractical for all but the smallest d.[3]

Template Optimization. Rather than densely covering the signal space, *template optimization* (TpopT) searches for a best matching template $\hat{\boldsymbol{s}}$, by numerically solving

$$\hat{\boldsymbol{s}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{s}\in S} f(\boldsymbol{s}) \equiv -\langle \boldsymbol{s}, \boldsymbol{x}\rangle \,. \tag{4.4}$$

The decision statistic is then $\hat{y}_{\mathrm{TpopT}}(\boldsymbol{x}) = 1 \iff \langle \hat{\boldsymbol{s}}(\boldsymbol{x}), \boldsymbol{x}\rangle \geq \tau$. Since the domain of optimization S is a Riemannian manifold, in principle, the optimization problem (4.4) can be solved by the Riemannian gradient iteration [198]

$$\boldsymbol{s}^{k+1} = \exp_{\boldsymbol{s}^k}\Big(-\alpha_k \,\mathrm{grad}[f](\boldsymbol{s}^k)\Big). \tag{4.5}$$

Here, k is the iteration index, $\exp_{\boldsymbol{s}}(\boldsymbol{v})$ is the exponential map at point $\boldsymbol{s}$, $\mathrm{grad}[f](\boldsymbol{s})$ is the Riemannian gradient[4] of the objective f at point $\boldsymbol{s}$, and α_k is the step size.

Alternatively, if the signal manifold S admits a global parameterization $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{\xi})$, we can optimize over the parameters $\boldsymbol{\xi}$, solving $\hat{\boldsymbol{\xi}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{\xi}} -\langle \boldsymbol{s}(\boldsymbol{\xi}), \boldsymbol{x}\rangle$ using the (Euclidean) gradient method:

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k + \alpha_k \cdot \Big(\nabla \boldsymbol{s}(\boldsymbol{\xi}^k)\Big)^{\mathrm{T}} \boldsymbol{x}, \tag{4.6}$$

where $\nabla \boldsymbol{s}(\boldsymbol{\xi}^k) \in \mathbb{R}^{D\times d}$ is the Jacobian matrix of $\boldsymbol{s}(\boldsymbol{\xi})$ at point $\boldsymbol{\xi}^k$. Finally, the estimated signal $\hat{\boldsymbol{s}}(\boldsymbol{x}) = \boldsymbol{s}(\hat{\boldsymbol{\xi}}(\boldsymbol{x}))$ and decision statistic $\hat{y}_{\mathrm{TpopT}}$ can be obtained from the estimated parameters $\hat{\boldsymbol{\xi}}$.

Of course, the optimization problem (4.4) is in general nonconvex, and methods (4.5)-(4.6)

[3] This inefficiency has motivated significant efforts in applied communities to optimize the placement of the templates $\boldsymbol{s}_i$, maximizing the statistical performance for a given fixed $n_{\mathrm{templates}}$ [27]. It is also possible to learn these templates from data, leveraging connections to neural networks [197]. Nevertheless, the curse of dimensionality remains in force.

[4] The Riemannian gradient is the projection of the Euclidean gradient $\nabla_{\boldsymbol{s}} f$ onto the tangent space $T_{\boldsymbol{s}} S$.

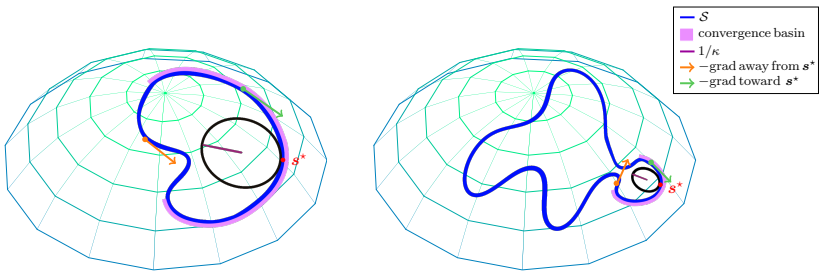

Figure 4.1: Relationship between curvature and convergence basins of gradient descent. Gradient descent has larger convergence basins under lower curvature (larger radius of osculating circle). Points within the convergence basin have gradient descent direction pointing "toward" $\boldsymbol{s}^\star$, while points outside the basin may have gradient descent pointing "away from" $\boldsymbol{s}^\star$.

only converge to global optima when they are initialized sufficiently close to the solution of (4.4). We can guarantee global optimality by employing multiple initializations $\boldsymbol{s}_1^0, \ldots, \boldsymbol{s}_{n_{\text{init}}}^0$, which cover the manifold S at some radius Δ where at least one initialization is guaranteed to produce a global optimizer.

In the next section, we will corroborate these intuitions with rigorous analysis. In subsequent sections, we will further develop more practical counterparts to (4.5)-(4.6) which (i) do not require an analytical representation of the signal manifold S [Section 4.4], and (ii) can be trained on sample data to improve statistical performance [Section 4.5].

4.3 Theory: Efficiency Gains over Matched Filtering

The efficiency advantage of optimization comes from its ability to use gradient information to rapidly converge to $\hat{\boldsymbol{s}} \approx \boldsymbol{s}_\natural$, within a basin of initializations $\boldsymbol{s}^0$ satisfying $d(\boldsymbol{s}^0, \boldsymbol{s}_\natural) \leq \Delta$: the larger the basin, the fewer initializations are needed to guarantee global optimality. The basin size Δ in turn depends on the geometry of the signal set S, through its *curvature*. Figure 4.1 illustrates the key intuition: if the curvature is small, there exists a relatively large region in which the gradient of the objective function points towards the global optimizer $\boldsymbol{s}^\star$. On the other hand, if the signal manifold is very curvy, there may only exist a relatively small region in which the gradient points in the correct direction.

We can formalize this intuition through the curvature of geodesics on the manifold S. For a smooth curve $\gamma : [0, T] \to S \subset \mathbb{R}^n$, with unit speed parameterization $\boldsymbol{\gamma}(t)$, $t \in [0, T]$, the

maximum curvature is

$$\kappa(\boldsymbol{\gamma}) = \sup_{t \in T} \|\ddot{\boldsymbol{\gamma}}(t)\|_2. \tag{4.7}$$

Geometrically, κ^{-1} is the minimum, over all points $\boldsymbol{\gamma}(t)$, of the radius of the osculating circle whose velocity and acceleration match those of $\boldsymbol{\gamma}$ at t. We extend this definition to S, a Riemannian submanifold of $\mathbb{R}^n$, by taking κ to be the maximum curvature of any geodesic on S:

$$\kappa(S) = \sup_{\boldsymbol{\gamma} \subset S \,:\, \text{unit-speed geodesic}} \kappa(\boldsymbol{\gamma}). \tag{4.8}$$

We call this quantity the *extrinsic geodesic curvature* of S.[5] Our main theoretical result shows that, as suggested by Figure 4.1 there is a $\Delta = 1/\kappa$ neighborhood within which gradient descent rapidly converges to a close approximation of $\boldsymbol{s}_\natural$:

Theorem 3. *Suppose the extrinsic geodesic curvature of S is bounded by κ. Consider the Riemannian gradient method* (4.5)*, with initialization satisfying $d(\boldsymbol{s}^0, \boldsymbol{s}_\natural) < 1/\kappa$, and step size $\tau = \frac{1}{64}$. Then when $\sigma \le c/(\kappa\sqrt{d})$, with high probability, we have for all k*

$$d(\boldsymbol{s}^{k+1}, \boldsymbol{s}_\natural) \le (1-\epsilon)\, d(\boldsymbol{s}^k, \boldsymbol{s}_\natural) + C\sigma\sqrt{d}. \tag{4.9}$$

Moreover, when $\sigma \le c/(\kappa\sqrt{D})$, with high probability, we have for all k

$$d(\boldsymbol{s}^k, \boldsymbol{s}^\star) \le C(1-\epsilon)^k \sqrt{f(\boldsymbol{s}^0) - f(\boldsymbol{s}^\star)}, \tag{4.10}$$

where $\boldsymbol{s}^\star$ is the unique minimizer of f over $B(\boldsymbol{s}_\natural, 1/\kappa)$. Here, C, c, ϵ are positive numerical constants.

Interpretation: Convergence to Optimal Statistical Precision In (4.9), we show that under a relatively mild condition on the noise, gradient descent exhibits linear convergence to a $\sigma\sqrt{d}$-

[5] Notice that $\kappa(S)$ measures how S curves *in the ambient space* $\mathbb{R}^n$; this is in contrast to traditional *intrinsic* curvature notions in Riemannian geometry, such as the sectional and Ricci curvatures. An extrinsic notion of curvature is relevant here because our objective function $f(\boldsymbol{s}) = -\langle \boldsymbol{s}, \boldsymbol{x} \rangle$ is defined extrinsically. Intrinsic curvature also plays an important role in our arguments — in particular, in controlling the effect of noise.

neighborhood of $\boldsymbol{s}_\natural$. This accuracy is the best achievable up to constants: for small σ, with high probability any minimizer $\boldsymbol{s}^\star$ satisfies $d(\boldsymbol{s}^\star, \boldsymbol{s}_\natural) > c\sigma\sqrt{d}$, and so the accuracy guaranteed by (4.9) is optimal up to constants. Also noteworthy is that both the accuracy and the required bound on the noise level σ are *dictated solely by the intrinsic dimension* d. The restriction $\sigma \leq c/(\kappa\sqrt{D})$ has a natural interpretation in terms of Figure 4.1 — at this scale, the noise "acts locally", ensuring that $\boldsymbol{s}^\star$ is close enough to $\boldsymbol{s}_\natural$ so that for any initialization in $B(\boldsymbol{s}_\natural, \Delta)$, the gradient points toward $\boldsymbol{s}^\star$. In (4.10) we also show that under a stronger condition on σ, gradient descent enjoys linear convergence for *all* iterations k.

Implications on Complexity. Here we compare the complexity required for MF and TpopT to achieve a target estimation accuracy $d(\hat{\boldsymbol{s}}, \boldsymbol{s}_\natural) \leq r$. The complexity of MF is simply N_r, the covering number of S with radius r. On the other hand, the complexity of TpopT is dictated by $n_{\text{init}} \times n_{\text{gradient-step}}$. We have $n_{\text{init}} = N_{1/\kappa}$ since TpopT requires initialization within radius $1/\kappa$ of $\boldsymbol{s}_\natural$, and $n_{\text{gradient-step}} \propto \log 1/\kappa r$ because gradient descent enjoys a linear convergence rate. Note that the above argument applies when $C\sigma d^{1/2}/\epsilon \leq r \leq 1/\kappa$, where the upper bound on r prescribes the regime where gradient descent is in action (otherwise TpopT and MF are identical), and the lower bound on r reflects the statistical limitation due to noise. Since the covering number $N_{\text{radius}} \propto (1/\text{radius})^d$, the complexities of the two methods T_{MF} and T_{TpopT} are given by

$$T_{\text{MF}} \propto 1/r^d, \qquad T_{\text{TpopT}} \propto \kappa^d \log(1/\kappa r). \tag{4.11}$$

Combining this with the range of r, it follows that TpopT always has superior dimensional scaling than MF whenever the allowable estimation error r is below $1/\kappa$ (and identical to MF above that). The advantage is more significant at lower noise and higher estimation accuracy.

Proof Ideas. The proof of Theorem 3 follows the intuition in Figure 4.1, by (i) considering a noiseless version of the problem and showing that in a $1/\kappa$ ball, the gradient points towards $\boldsymbol{s}_\natural$, and (ii) controlling the effect of noise, by bounding the maximum component $T^{\max}$ of the noise

z along any tangent vector $v \in T_s S$ at any point $s \in B(s_\natural, \Delta)$. By carefully controlling $T^{\max}$, we are able to achieve rates driven by intrinsic dimension, not ambient dimension. Intuitively, this is because the collection of tangent vectors, i.e., the tangent bundle, has dimension $2d$. Our proof involves a discretization argument, which uses elements of Riemannian geometry (Toponogov's theorem on geodesic triangles, control of parallel transport via the second fundamental form [199, 200]). To show convergence of iterates (4.10), we show that in a $1/\kappa$ region, the objective f enjoys Riemannian strong convexity and Lipschitz gradients [198]. Please see the supplementary material for complete proofs.

4.4 Nonparametric TpopT via Embedding and Kernel Interpolation

The theoretical results in Section 4.3 rigorously quantify the advantages of TpopT in detecting and estimating signals from low-dimensional families. A straightforward application of TpopT requires a precise analytical characterization of the signal manifold. In this section, we develop more practical, nonparametric extension of TpopT, which is applicable in scenarios in which we *only* have examples $s_1, \ldots, s_N$ from S. This extension will maintain the test-time efficiency advantages of TpopT.

Embedding. We begin by embedding the example points $s_1, \ldots, s_N \in \mathbb{R}^n$ into a lower-dimensional space $\mathbb{R}^d$, producing data points $\xi_1, \ldots, \xi_N \in \mathbb{R}^d$. The mapping φ should preserve pairwise distances and can be chosen in a variety of ways; Because the classical Multidimensional Scaling (MDS) setup on Euclidean distances is equivalent to Principal Component Analysis (PCA), we simply use PCA in our experiments. Assuming that φ is bijective over S, we can take $s = s(\xi)$ as an approximate parameterization of S, and develop an optimization method which, given an input x, searches for a parameter $\xi \in \mathbb{R}^d$ that minimizes $f(s(\xi)) = -\langle s(\xi), x \rangle$.

Kernel Interpolated Jacobian Estimates. In the nonparameteric setting, we only know the values of $s(\xi)$ at the finite point set $\xi_1, \ldots, \xi_N$, and we do not have any direct knowledge of the functional form of the mapping $s(\cdot)$ or its derivatives. To extend TpopT to this setting, we can

estimate the Jacobian $\nabla \boldsymbol{s}(\boldsymbol{\xi})$ at point $\boldsymbol{\xi}_i$ by solving a weighted least squares problem

$$\widehat{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_i) = \arg\min_{\boldsymbol{J} \in \mathbb{R}^{D \times d}} \sum_{j=1}^{N} w_{j,i} \left\| \boldsymbol{s}_j - \boldsymbol{s}_i - \boldsymbol{J}(\boldsymbol{\xi}_j - \boldsymbol{\xi}_i) \right\|_2^2, \tag{4.12}$$

where the weights $w_{j,i} = \Theta(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ are generated by an appropriately chosen kernel Θ. The least squares problem (4.12) is solvable in closed form. In practice, we prefer compactly supported kernels, so that the sum in (4.12) involves only a small subset of the points $\boldsymbol{\xi}_j$;[6] in experiment, we choose Θ to be a truncated radial basis function kernel $\Theta_{\lambda,\delta}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\lambda \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2) \cdot \mathbb{1}_{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 < \delta}$. When example points $\boldsymbol{s}_i$ are sufficiently dense and the kernel Θ is sufficiently localized, $\widehat{\nabla \boldsymbol{s}}(\boldsymbol{\xi})$ will accurately approximate the true Jacobian $\nabla \boldsymbol{s}(\boldsymbol{\xi})$.

Expanding the Basin of Attraction using Smoothing. In actual applications such as computer vision and astronomy, the signal manifold S often exhibits large curvature κ, leading to a small basin of attraction. One classical heuristic for increasing the basin size is to *smooth* the objective function f. We can incorporate smoothing by taking gradient steps with a kernel smoothed Jacobian,

$$\widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_i) = Z^{-1} \sum_j w_{j,i}\, \widehat{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_j), \tag{4.13}$$

where $w_{j,i} = \Theta_{\lambda_s, \delta_s}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ and $Z = \sum_j w_{j,i}$. The gradient iteration becomes

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k + \alpha_k \widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi}^k)^{\mathrm{T}} \boldsymbol{x}. \tag{4.14}$$

When the Jacobian estimate $\widehat{\nabla \boldsymbol{s}}(\boldsymbol{\xi})$ accurately approximates $\nabla \boldsymbol{s}(\boldsymbol{\xi})$, we have

$$\widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_i)^{\mathrm{T}} \boldsymbol{x} \approx Z^{-1} \sum_j w_{j,i} \nabla \boldsymbol{s}(\boldsymbol{\xi}_j)^{\mathrm{T}} \boldsymbol{x} = \nabla \left[Z^{-1} \sum_j w_{j,i} f(\boldsymbol{s}(\boldsymbol{\xi}_j)) \right]. \tag{4.15}$$

i.e., $\widetilde{\nabla \boldsymbol{s}}^{\,\mathrm{T}}$ is an approximate gradient for a smoothed version $\widetilde{f}$ of the objective f. Figure 4.2 illustrates smoothed optimization landscapes $\widetilde{f}$ for different levels of smoothing, i.e., different

[6] In our experiments on gravitational wave astronomy, we introduce an additional quantization step, computing approximate Jacobians on a regular grid $\hat{\xi}_1, \ldots, \hat{\xi}_{N'}$ of points in the parameter space Ξ.

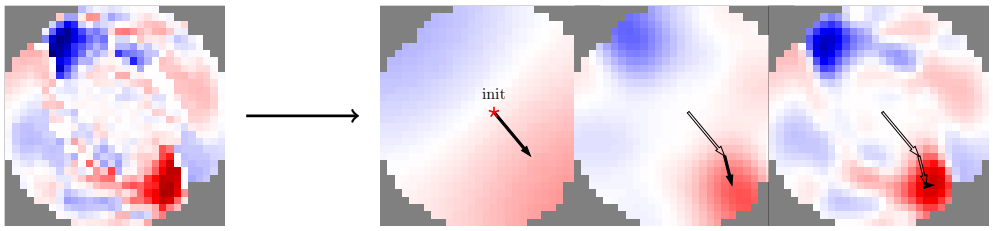

Figure 4.2: Illustration of 2-dim signal embeddings and the parameter optimization procedure for gravitational wave signals.

choices of λ_s. In general, the more smoothing is applied, the broader the basin of attraction. We employ a coarse-to-fine approach, which starts with a highly smoothed landscape (small λ_s) in the first iteration and decreases the level of smoothing from iteration to iteration — see Figure 4.2.

These observations are in line with theory: because our embedding approximately preserves Euclidean distances, $\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2 \approx \|\boldsymbol{s}_i - \boldsymbol{s}_j\|_2$, we have

$$\widetilde{f}(\boldsymbol{s}(\boldsymbol{\xi}_i)) = Z^{-1} \sum_j \Theta(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \left\langle \boldsymbol{s}_j, \boldsymbol{x} \right\rangle \approx \langle Z^{-1} \sum_j \Theta(\boldsymbol{s}_i, \boldsymbol{s}_j) \boldsymbol{s}_j, \boldsymbol{x} \rangle, \tag{4.16}$$

i.e., applying kernel smoothing in the parameter space is nearly equivalent to applying kernel smoothing to the signal manifold S. This smoothing operation expands the basin of attraction $\Delta = 1/\kappa$, by reducing the manifold curvature κ. Empirically, we find that with appropriate smoothing often a single initialization suffices for convergence to global optimality, suggesting this as a potential key to breaking the curse of dimensionality.

4.5 Training Nonparametric TpopT

In the section above, we described nonparametric TpopT for finding the matching template by the iterative gradient solver (4.5). Note that this framework requires pre-computing the Jacobians $\nabla \boldsymbol{s}(\boldsymbol{\xi})$ and determining optimization hyperparameters, including the step sizes α_k and kernel width parameters λ_k at each layer. In this section, we adapt TpopT into a trainable architecture, which essentially learns all the above quantities from data to further improve performance.

Recall the gradient descent iteration (4.14) in TpopT. Notice that if we define a collection of

matrices $\boldsymbol{W}(\boldsymbol{\xi}_i, k) = \alpha_k \widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_i)^{\mathrm{T}} \in \mathbb{R}^{d \times D}$ indexed by $\boldsymbol{\xi}_i \in \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N\}$ and $k \in \{1, \ldots, K\}$ where K is the total number of iterations, then the iteration can be rewritten as

$$\boldsymbol{\xi}^{k+1} = C^{-1} \sum_{i=1}^{N} w_{k,i} \left(\boldsymbol{\xi}_i + \boldsymbol{W}(\boldsymbol{\xi}_i, k)\, \boldsymbol{x} \right), \tag{4.17}$$

where $w_{k,i} = \Theta_{\lambda_k, \delta_k}(\boldsymbol{\xi}^k, \boldsymbol{\xi}_i)$, and $C = \sum_i w_{k,i}$. Equation (4.17) can be interpreted as a kernel interpolated gradient step, where the $\boldsymbol{W}$ matrices summarize the Jacobian and step size information. Because Θ is compactly supported, this sum involves only a small subset of the sample points $\boldsymbol{\xi}_i$. Now, if we "unroll' the optimization by viewing each gradient descent iteration as one layer of a trainable network, we arrive at a trainable TpopT architecture, as illustrated in Figure 4.3. Here the trainable parameters in the network are the $\boldsymbol{W}(\boldsymbol{\xi}_i, k)$ matrices and the kernel width parameters λ_k.

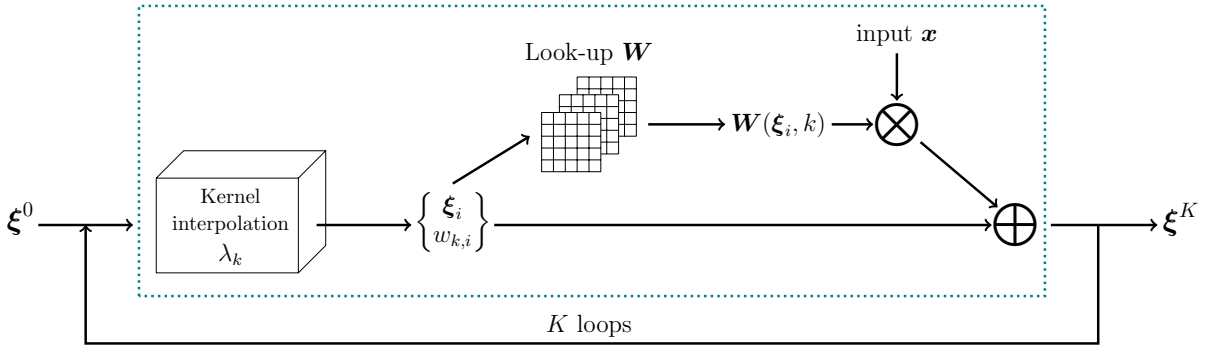

Figure 4.3: Architecture of trainable TpopT. The model takes $\boldsymbol{x}$ as input and starts with a fixed initialization $\boldsymbol{\xi}^0$, and outputs $\boldsymbol{\xi}^K$ after going through K layers. The trainable parameters are the collection of $\boldsymbol{W}(\boldsymbol{\xi}_i, k)$ matrices and kernel width parameters λ_k.

Following our heuristic that the $\boldsymbol{W}(\boldsymbol{\xi}_i, k)$ matrices were originally the combination of Jacobian and step size, we can initialize these matrices as $\alpha_k \widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi}_i)^{\mathrm{T}}$. For the loss function during training, we use the square loss between the network output $\boldsymbol{\xi}^K(\boldsymbol{x})$ and the optimal quantization point $\boldsymbol{\xi}^*(\boldsymbol{x}) = \arg\max_{i=1,\ldots,N} \left\langle \boldsymbol{s}(\boldsymbol{\xi}_i), \boldsymbol{x} \right\rangle$, namely

$$L = \frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \|\boldsymbol{\xi}^K(\boldsymbol{x}_j) - \boldsymbol{\xi}^*(\boldsymbol{x}_j)\|_2^2 \tag{4.18}$$

for a training set $\{\boldsymbol{x}_j\}_{j=1}^{N_{\text{train}}}$ with positively-labeled data only. This loss function is well-aligned with the signal estimation task, and is also applicable to detection.

In summary, the trainable TpopT architecture consists of the following steps:

- Create embeddings $\boldsymbol{s}_i \mapsto \boldsymbol{\xi}_i$.
- Estimate Jacobians $\nabla \boldsymbol{s}(\boldsymbol{\xi})$ at points $\boldsymbol{\xi}_i$ by weighted least squares.
- Estimate smoothed Jacobians $\widetilde{\nabla \boldsymbol{s}}(\boldsymbol{\xi})$ at any $\boldsymbol{\xi}$ by kernel smoothing.
- Select a multi-level smoothing scheme.
- Train the model with unrolled optimization.

4.6 Experiments

We apply the trainable TpopT to gravitational wave detection, where MF is the current method of choice, and show a significant improvement in efficiency-accuracy tradeoffs. We further demonstrate its wide applicability on low-dimensional data with experiments on handwritten digit data.

To compare the efficiency-accuracy tradeoffs of MF and TpopT models, we note that (i) for MF, the computation cost of the statistic $\max_{i=1,\ldots,n} \langle \boldsymbol{s}_i, \boldsymbol{x} \rangle$ is dominated by the cost of n length D inner products, requiring nD multiplication operations. For TpopT, with M parallel initializations, K iterations of the gradient descent (4.17), m neighbors in the truncated kernel, and a final evaluation of the statistic, we require $MD(Kdm + 1)$ multiplications; other operations including the kernel interpolation and look-up of pre-computed gradients have negligible test-time cost.

4.6.1 Gravitational Wave Detection

We aim to detect a family of gravitational wave signals in Gaussian noise. Each gravitational wave signal is a one-dimensional chirp-like signal – see Figure 4.4 (left).[7] Please refer to section B.5 in the appendix for data generation details.

Based on their physical modeling, gravitational wave signals are equipped with a set of physical parameters, such as the masses and three-dimensional spins of the binary black holes that generate them, etc. While it is tempting to directly optimize on this native parameter space, unfortunately

[7]The raw data of gravitational wave detection is a noisy one-dimensional time series, where gravitational wave signals can occur at arbitrary locations. We simplify the problem by considering input segments of fixed time duration.

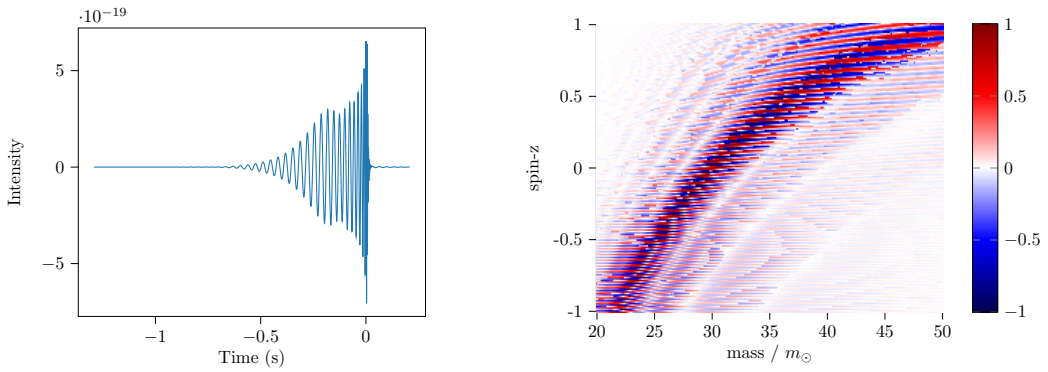

Figure 4.4: **Left:** Example of a gravitational wave signal. **Right:** Optimization landscape in the physical parameter space (mass-spin-z), shown as the heatmap of signal correlations.

the optimization landscape on this space turns out to be rather unfavorable, as shown in Figure 4.4 (right). We see that the objective function has many spurious local optimizers and is poorly conditioned. Therefore, we still resort to signal embedding to create an alternative set of approximate "parameters" that are better suited for optimization.

For the signal embedding, we apply PCA with dimension 2 on a separate set of 30,000 noiseless waveforms drawn from the same distribution. Because the embedding dimension is relatively low, here we quantize the embedding parameter space with an evenly-spaced grid, with the range of each dimension evenly divided into 30 intervals. The value $\boldsymbol{\xi}^0$ at the initial layer of TpopT is fixed at the center of this quantization grid. Prior to training, we first determine the optimization hyperparameters (step sizes and smoothing levels) using a layer-wise greedy grid search, where we sequentially choose the step size and smoothing level at each layer as if it were the final layer. This greedy approach significantly reduces the cost of the search. From there, we use these optimization hyperparameters to initialize the trainable TpopT network, and train the parameters on the training set. We use the Adam [152] optimizer with batch size 1000 and constant learning rate 10^{-2}. Regarding the computational cost of TpopT, we have $M = 1$, $d = 2$, $m = 4$ during training and $m = 1$ during testing. The test time complexity of K-layer TpopT is $D(2K+1)$.

To evaluate the performance of matched filtering at any given complexity m, we randomly generate 1,000 independent sets of m templates drawn from the above distribution, evaluate the ROC curves of each set of templates on the validation set, and select the set with the highest area-under-curve (AUC) score. This selected template bank is then compared with TpopT on the shared

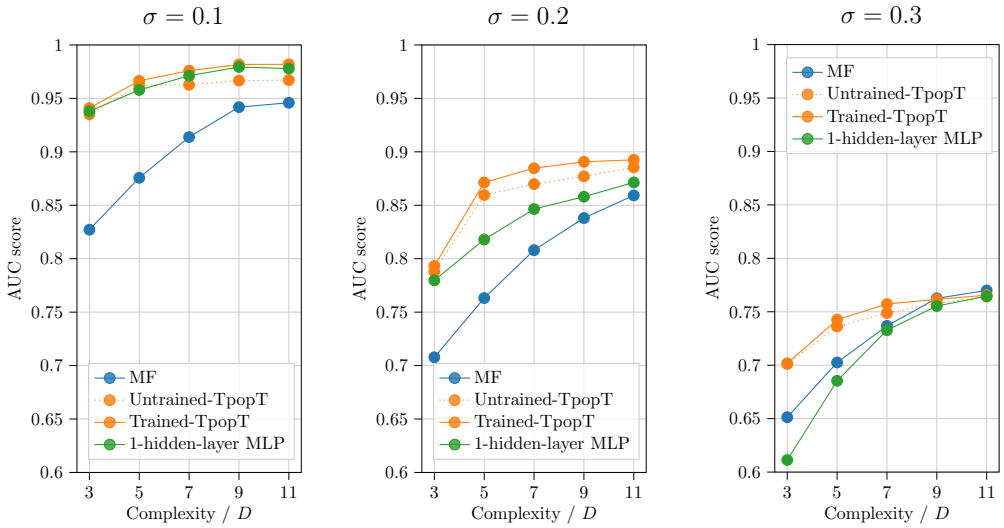

Figure 4.5: This figure compares the performance of four methods: (1) matched filtering (MF), (2) Template optimization (TpopT) without training, (3) TpopT with training, and (4) multi-layer perceptron (MLP) with one hidden layer. All methods are compared at three noise levels. We see that TpopT performs well in low to moderate noise, which matches theoretical results.

test set.

Figure 4.6.1 shows the comparison of efficiency-accuracy trade-offs for this task between matched filtering and TpopT after training. We see that TpopT achieves significantly higher detection accuracy compared with MF at equal complexity. At low to moderate noise levels, Trained-TpopT performs the best, followed by MLP, and matched filtering performs the worst. As noise level increases, MLP's performance worsens most significantly, becoming the worst at $\sigma = 0.3$.

4.6.2 Handwritten Digit Recognition

In this second experiment, we apply TpopT to the classic task of handwritten digit recognition using the MNIST [201] dataset, in particular detecting the digit 3 from all other digits. We apply random Euclidean transformations to all images, with translation uniformly between ±0.1 image size on both dimensions and rotation angle uniformly between ±30°.

Since the signal space here is nonparametric, we first create a 3-dimensional PCA embedding from the training set, and Figure 4.6 (left) shows a slice of the embedding projected onto the first two embedding dimensions. See supplementary for experiment details. Regarding the computa-

tional cost of TpopT, we have $M = 1$, $d = 3$, $m = 5$ during training and $m = 1$ during testing. Since the complexity is measured at test time, the complexity with K-layer TpopT is $D(3K + 1)$. Additional experimental details can be found in B.5.

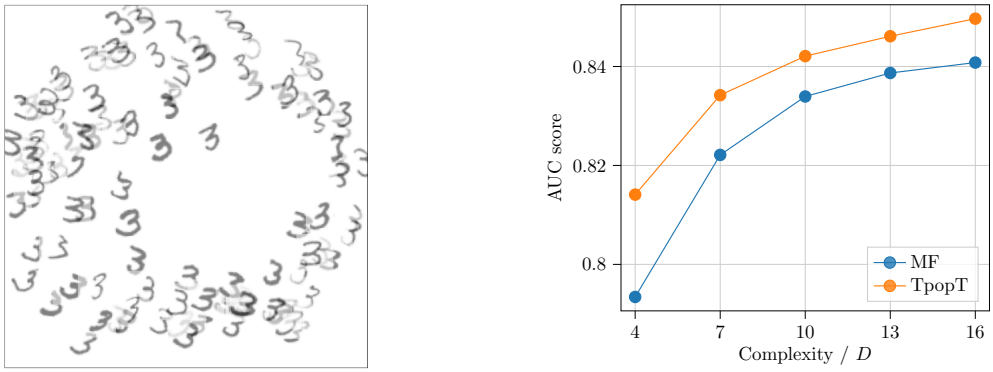

Figure 4.6: **Left:** A slice of the 3-d embeddings projected onto the first two dimensions. **Right:** Classification scores of MF and TpopT at different complexity levels, for handwritten digit recognition.

Matched filtering is also evaluated similarly as in the previous experiment. We first set aside a random subset of 500 images of digit 3 from the MNIST training set and construct the validation set from it. The remaining images are used to randomly generate 1,000 independent sets of transformed digits 3, and the best-performing set of templates on the validation set is selected as the MF template bank, and compared with TpopT on the shared test set. Figure 4.6 (right) shows the comparison of efficiency-accuracy tradeoffs between the two methods, and we see a consistently higher detection accuracy of trained TpopT over MF at equal complexities.

4.7 Discussion and Limitations

In this section, we studied TpopT as an approach to efficient detection of low-dimensional signals. We provided a proof of convergence of Riemannian gradient descent on the signal manifold, and demonstrated its superior dimension scaling compared to MF. We also proposed the trainable TpopT architecture that can handle general nonparametric families of signals. Experimental results show that trained TpopT achieves significantly improved efficiency-accuracy tradeoffs than MF, especially in the gravitational wave detection task where MF is the method of choice.

The principal limitation of nonparametric TpopT is its storage complexity: it represents the manifold using a dense collection of points and Jacobians, with cost exponential in intrinsic dimension d. At the same time, we note that the same exponential storage complexity is encountered by matched filtering with a pre-designed template bank. In some sense, this exponential resource requirement reflects an intrinsic constraint of the signal detection problem, unless more structures within the signal space can be exploited. Both TpopT and its nonparametric extension achieve exponential improvements in test-time efficiency compared to MF; nevertheless, our theoretical results retain an exponential dependence on intrinsic dimension d, due to the need for multiple initializations. In experiments, the proposed smoothing allows convergence to global optimality from a single initialization. Our current theory does not fully explain this observation; this is an important direction for future work.

An advantage of MF not highlighted in the discussion is its efficiency in handling noisy time series, using the fast Fourier transform. This enables MF to rapidly locate signals that occur at a-priori unknown spatial/temporal locations. Developing a convolutional version of TpopT with similar advantages is another important direction.

Finally, our gravitational wave experiments use synthetic data with known ground truth, in order to corroborate the key messages of this work. Future experiments that explore broader and more realistic setups will be an important empirical validation of the proposed method.

Chapter 5: Conclusion

As the fields of machine learning and signal processing continue to advance, their methodologies will be deployed to increasingly more sophisticated and large-scale tasks. This in turn poses increasingly high demands for model efficiency and interpretability, which are critical to processing enormous volumes of data and extracting meaningful insights from it. In this thesis, we have focused on the problem of detecting and estimating parametric families of signals, a fundamental problem in classic and modern applications alike, and proposed a series of interpretable trainable architectures for both statistical accuracy and computational efficiency. Although plenty of works exist on applying deep learning models to gravitational wave detection, we take the different approach of bridging classic statistics and modern learning, achieving "the best of both worlds".

In Chapter 1, we motivated the need for interpretable learning architectures, and presented an overview of the classic problem of detecting a parametric family of signals, as well as the matched filtering method and its limitations in computational efficiency. We also introduced the application of gravitational wave detection, where accuracy, efficiency and interpretability are three prominent desiderata.

In Chapter 2, we first illustrated the intrinsic suboptimality of matched filtering for parametric signal detection, and then proposed the MNet architectures as possible remedies. MNet-Shallow incorporates trainable templates, leading to a strict performance gain. MNet-Deep further gets rid of the convex boundary constraint of the matched filtering architecture by using a deep ReLU network, thereby able to better handle non-Gaussian noise distributions, as is the case in many realistic applications.

In Chapter 3, we turned our focus to computational efficiency of the models, and proposed to combine the MNet-Shallow architecture with hierarchical decision rules, leading to a trainable hierarchical architecture. Taking advantage of the class imbalance structure in tasks such

as gravitational wave detection, the progressive rejection scheme is able to significantly reduce computational cost while maintaining similar accuracy.

In Chapter 4, we continued our focus on computational efficiency, but utilizing a different type of structure in the problem — the geometric structure of the signal set, and centered on the core idea that optimization over the signal space is more efficiency than covering. This leads to the template optimization (TpopT) method for detection and estimation, which we show theoretically to be exponentially more efficient than template covering. To unlock the full power of the TpopT model, we reconfigure the architecture to be trainable through unrolled optimization and kernel interpolation. Combined, the proposed trainable TpopT enjoys a significant performance-accuracy trade-off improvement over matched filtering, and remains highly interpretable compared with many other deep learning architectures.

While the majority of our empirical evaluations focus on the task of gravitational wave detection, our proposed methods are broadly applicable to any task with a signal family of interest, as shown by the handwritten digit experiment for TpopT. Indeed, in virtually any scenario where the conceptual idea of template matching is present, from geophysics and radar to modern learning architecture that utilize template banks, it is possible to incorporate training and unrolled optimization for an interpretable architecture with improved in statistical accuracy and computational efficiency.

5.1 Limitations and Future Work

Despite the work present in this thesis, there remain many limitations and possibilities for further improvements. We will outline a few in this section.

Throughout the discussion, we have approached the signal detection problem such that the signal appearance location is determined in advance, and that the only uncertainty lies in the signal itself. We first note that this is actually not as unrealistic as it may sound. When the temporal or spatial uncertainty is taken into account, one possible solution is to simply apply the fixed-location model in a sliding-window fashion over all possible locations. This can well be feasible

on relatively small scales, similar to what is done for small convolutional kernels in convolutional networks. At larger scales, the efficiency of frequency domain processing through fast Fourier transforms begins to appear, and matched filtering is naturally compatible with such processing, giving it an advantage that is not present in fixed-location scenarios. Nonetheless, both MNet architectures and the hierarchical network architecture share the same frequency domain compatibility, so the accuracy-efficiency comparison is still valid. The only genuine concern lies with the TpopT architecture, where the optimization configuration removes the affinity with frequency domain processing. One promising solution is to include the temporal/spatial dimension as part of a joint optimization, which can also expand the model to accommodate multiple different signals at various locations. Such extensions of the model also has strong connections to the deconvolution literature [202], and will be an important direction for future work.

Another natural question is whether one can combine the two types of structures leveraged in Chapters 3 and 4 — the problem structure and the geometric structure, and produce a unified framework. For instance, one can conceive an architecture that applies a rejection threshold after each iteration of the unrolled optimization. In addition to these aforementioned structures, in practical applications there can exist more problem-specific structures on the dataset, such as time-frequency domain concentration for gravitational wave signals. Due to their problem-specific nature, it might be potentially more challenging to design general-purpose architectures for them, but studying ways to incorporate such information are promising in further boosting detection accuracy and efficiency.

Regarding the applicability of the proposed architectures, as previously discussed, they are by no means restricted to the gravitational wave application. Nonetheless, showcasing concrete improvements over current methods in some other realistic problem setups, such as neuroscience, geophysics or radar, will further demonstrate their applicability. At the same time, the proposed ideas can potentially find applications in modern learning pipelines as well, serving as a plug-in module whenever a similarity search is conducted. The unrolled architecture enables uninterrupted gradient flow, and allows end-to-end training with other learning modules. Exploring such poten-

tial in modern deep learning will also be an important future direction. Furthermore, while our focus has been on signal detection and estimation, similar ideas can potentially be applicable to signal sensing too. For example, there exist sensing methods in electrochemical microscopy that utilize compressed sensing techniques and can be used for detection of different template shapes [203, 204]. In such circumstances, an unrolled optimization solver with trainable templates will be able to further improve the model efficiency.

There are many other aspects of the problem not covered by the thesis, such as better understanding the sample complexity required for detection, and online learning with time-varying families of signals. Answering these questions will provide a more holistic understanding for this fundamental problem.

Finally, we sincerely hope that the ideas, methods and results presented in this thesis can serve the scientific and engineering community in developing more efficient data processing pipelines, deepening our understanding of trainable models in this era of big data, and revealing more secrets of the universe.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[7] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[8] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[9] O. Vinyals *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[10] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[11] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *Advances in neural information processing systems*, vol. 34, pp. 21 696–21 707, 2021.

[12] S. Buchanan, D. Gilboa, and J. Wright, *Deep networks and the multiple manifold problem*, 2020. arXiv: `2008.11245 [stat.ML]`.

[13] T. Wang, S. Buchanan, D. Gilboa, and J. Wright, "Deep networks provably classify data on curves," *Advances in neural information processing systems*, vol. 34, pp. 28 940–28 953, 2021.

[14] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[15] M. Craven and J. Shavlik, "Extracting tree-structured representations of trained networks," *Advances in neural information processing systems*, vol. 8, 1995.

[16] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3429–3437.

[17] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, ""' why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[20] J. D. Angrist and J.-S. Pischke, *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2009.

[21] P. D. Fajgelbaum, P. K. Goldberg, P. J. Kennedy, and A. K. Khandelwal, "The return to protectionism," *The Quarterly Journal of Economics*, vol. 135, no. 1, pp. 1–55, 2020.

[22] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: Methodological failures and recommendations for the future," *NPJ digital medicine*, vol. 5, no. 1, p. 48, 2022.

[23] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on machine learning*, 2010, pp. 399–406.

[24] M. Andrychowicz *et al.*, "Learning to learn by gradient descent by gradient descent," *Advances in neural information processing systems*, vol. 29, 2016.

[25] O. Wichrowska *et al.*, "Learned optimizers that scale and generalize," in *International conference on machine learning*, PMLR, 2017, pp. 3751–3760.

[26] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[27] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," *Physical Review D*, vol. 60, no. 2, p. 022 002, 1999.

[28] Y. Shi, Z. Nenadic, and X. Xu, "Novel use of matched filtering for synaptic event detection and extraction," *PLoS One*, vol. 5, no. 11, e15517, 2010.

[29] E. Caffagni, D. W. Eaton, J. P. Jones, and M. van der Baan, "Detection and analysis of microseismic events using a matched filtering algorithm (mfa)," *Geophysical Journal International*, vol. 206, no. 1, pp. 644–658, 2016.

[30] B. Rousset *et al.*, "A geodetic matched filter search for slow slip with application to the mexico subduction zone," *Journal of Geophysical Research: Solid Earth*, vol. 122, no. 12, pp. 10–498, 2017.

[31] K. Picos, V. H. Diaz-Ramirez, V. Kober, A. S. Montemayor, and J. J. Pantrigo, "Accurate three-dimensional pose recognition from monocular images using template matched filtering," *Optical Engineering*, vol. 55, no. 6, pp. 063 102–063 102, 2016.

[32] T. Pardhu, A. K. Sree, and K Tanuja, "Design of matched filter for radar applications," *Electrical and Electronics Engineering: An International Journal (ELELIJ) Vol*, vol. 3, 2014.

[33] D. Johnson, "Complex scatterer reconstruction using multistatic spherical wave isar fourier template matching," in *2009 International Conference on Electromagnetics in Advanced Applications*, IEEE, 2009, pp. 291–294.

[34] T. S. Murphy, M. J. Holzinger, and B. Flewelling, "Space object detection in images using matched filter bank and bayesian update," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 3, pp. 497–509, 2017.

[35] K. Sohn and H. Lee, "Learning invariant representations with local transformations," *arXiv preprint arXiv:1206.6418*, 2012.

[36] A. Kanazawa, A. Sharma, and D. Jacobs, "Locally scale-invariant convolutional neural networks," *arXiv preprint arXiv:1412.5104*, 2014.

[37] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Oriented response networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 519–528.

[38] LIGO and Virgo Collaborations, "Observation of gravitational waves from a binary black hole merger," *Phys. Rev. Lett.*, vol. 116, p. 061 102, 6 Feb. 2016.

[39] B. P. Abbott and et al., "Gwtc-1: A gravitational-wave transient catalog of compact binary mergers observed by ligo and virgo during the first and second observing runs," *Physical Review X*, vol. 9, no. 3, Sep. 2019.

[40] V. LSC and KAGRA, "Gwtc-2: Compact binary coalescences observed by ligo and virgo during the first half of the third observing run," *arXiv e-prints*, arXiv:2010.14527, arXiv:2010.14527, Oct. 2020. arXiv: 2010.14527 [gr-qc].

[41] B. P. Abbott and et al., "Gw170817: Observation of gravitational waves from a binary neutron star inspiral," *Phys. Rev. Lett.*, vol. 119, p. 161 101, 16 Oct. 2017.

[42] B. P. Abbott and et al., "Multi-messenger observations of a binary neutron star merger," *Astrophysical Journal Letters*, vol. 848, no. 2, L12, p. L12, Oct. 2017. arXiv: 1710.05833 [astro-ph.HE].

[43] T. Akutsu *et al.*, "Overview of KAGRA: Detector design and construction history," *Progress of Theoretical and Experimental Physics*, Aug. 2020, ptaa125. eprint: https://academic.oup.com/ptep/advance-article-pdf/doi/10.1093/ptep/ptaa125/34386189/ptaa125.pdf.

[44] K. L. Dooley *et al.*, "GEO 600 and the GEO-HF upgrade program: Successes and challenges," *Classical and Quantum Gravity*, vol. 33, no. 7, p. 075 009, Mar. 2016.

[45] F. e. a. Acernese, "Advanced Virgo: a second-generation interferometric gravitational wave detector," *Classical and Quantum Gravity*, vol. 32, no. 2, 024001, p. 024 001, Jan. 2015. arXiv: 1408.3978 [gr-qc].

[46] A. Abramovici *et al.*, "LIGO: The Laser Interferometer Gravitational-Wave Observatory," *Science*, vol. 256, no. 5055, pp. 325–333, Apr. 1992.

[47] LIGO Scientific Collaboration, "Advanced LIGO," *Classical and Quantum Gravity*, vol. 32, no. 7, 074001, p. 074 001, Apr. 2015. arXiv: 1411.4547 [gr-qc].

[48] C Affeldt *et al.*, "Advanced techniques in GEO 600," *Classical and Quantum Gravity*, vol. 31, no. 22, p. 224 002, Nov. 2014.

[49] F. Acernese *et al.*, "Increasing the astrophysical reach of the advanced virgo detector via the application of squeezed vacuum states of light," *Phys. Rev. Lett.*, vol. 123, p. 231 108, 23 Dec. 2019.

[50] M. Tse *et al.*, "Quantum-enhanced advanced ligo detectors in the era of gravitational-wave astronomy," *Phys. Rev. Lett.*, vol. 123, p. 231 107, 23 Dec. 2019.

[51] A. Einstein, "Näherungsweise Integration der Feldgleichungen der Gravitation," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin*, pp. 688–696, Jan. 1916.

[52] A. Einstein, "Über Gravitationswellen," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin*, pp. 154–167, Jan. 1918.

[53] L. S. Finn, "Detection, measurement, and gravitational radiation," *Phys. Rev. D*, vol. 46, pp. 5236–5249, 12 Dec. 1992.

[54] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," Phys. Rev. D, vol. 60, no. 2, 022002, p. 022 002, Jul. 1999. arXiv: gr-qc/9808076 [gr-qc].

[55] L. S. Finn, "Aperture synthesis for gravitational-wave data analysis: Deterministic sources," Phys. Rev. D, vol. 63, no. 10, 102001, p. 102 001, May 2001. arXiv: gr-qc/0010033 [gr-qc].

[56] W. G. Anderson, P. R. Brady, J. D. Creighton, and É. É. Flanagan, "Excess power statistic for detection of burst sources of gravitational radiation," Phys. Rev. D, vol. 63, no. 4, 042003, p. 042 003, Feb. 2001. arXiv: gr-qc/0008066 [gr-qc].

[57] S. Klimenko and G. Mitselmakher, "A wavelet method for detection of gravitational wave bursts," *Classical and Quantum Gravity*, vol. 21, no. 20, S1819–S1830, Oct. 2004.

[58] W. G. Anderson, P. R. Brady, J. D. E. Creighton, and É. É. Flanagan, "A Power Filter for the Detection of Burst Sources of Gravitational Radiation in Interferometric Detectors," *International Journal of Modern Physics D*, vol. 9, no. 3, pp. 303–307, Jan. 2000. arXiv: gr-qc/0001044 [gr-qc].

[59] S. W. Hawking and W. Israel, *Three Hundred Years of Gravitation*. 1989.

[60] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," *Phys. Rev. D*, vol. 60, p. 022 002, 2 Jun. 1999.

[61] C. Cutler *et al.*, "The last three minutes: Issues in gravitational-wave measurements of coalescing compact binaries," Phys. Rev. Lett., vol. 70, no. 20, pp. 2984–2987, May 1993. arXiv: astro-ph/9208005 [astro-ph].

[62] C. Cutler and É. E. Flanagan, "Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform\?" Phys. Rev. D, vol. 49, no. 6, pp. 2658–2697, Mar. 1994. arXiv: gr-qc/9402014 [gr-qc].

[63] É. É. Flanagan and S. A. Hughes, "Measuring gravitational waves from binary black hole coalescences. I. Signal to noise for inspiral, merger, and ringdown," Phys. Rev. D, vol. 57, no. 8, pp. 4535–4565, Apr. 1998. arXiv: gr-qc/9701039 [gr-qc].

[64] É. É. Flanagan and S. A. Hughes, "Measuring gravitational waves from binary black hole coalescences. II. The waves' information and its extraction, with and without templates," Phys. Rev. D, vol. 57, no. 8, pp. 4566–4587, Apr. 1998. arXiv: gr-qc/9710129 [gr-qc].

[65] B. e. a. Abbott, "Analysis of LIGO data for gravitational waves from binary neutron stars," Phys. Rev. D, vol. 69, no. 12, 122001, p. 122 001, Jun. 2004. arXiv: gr-qc/0308069 [gr-qc].

[66] *Pycbc software releases*, https://github.com/gwastro/pycbc/releases.

[67] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, "FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries," Phys. Rev. D, vol. 85, no. 12, 122006, p. 122 006, Jun. 2012. arXiv: gr-qc/0509116 [gr-qc].

[68] C. M. Biwer *et al.*, "PyCBC Inference: A Python-based Parameter Estimation Toolkit for Compact Binary Coalescence Signal," PASP, vol. 131, no. 996, p. 024 503, Feb. 2019. arXiv: 1807.10312 [astro-ph.IM].

[69] B. Allen, "χ^2 time-frequency discriminator for gravitational wave detection," Phys. Rev. D, vol. 71, no. 6, 062001, p. 062 001, Mar. 2005. arXiv: gr-qc/0405045 [gr-qc].

[70] T. Dal Canton *et al.*, "Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors," Phys. Rev. D, vol. 90, no. 8, 082004, p. 082 004, Oct. 2014. arXiv: 1405.6731 [gr-qc].

[71] S. A. Usman *et al.*, "The PyCBC search for gravitational waves from compact binary coalescence," *Classical and Quantum Gravity*, vol. 33, no. 21, 215004, p. 215 004, Nov. 2016. arXiv: 1508.02357 [gr-qc].

[72] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, "Rapid detection of gravitational waves from compact binary mergers with PyCBC Live," Phys. Rev. D, vol. 98, no. 2, 024050, p. 024 050, Jul. 2018. arXiv: 1805.11174 [gr-qc].

[73] C. Messick *et al.*, "Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data," Phys. Rev. D, vol. 95, no. 4, 042001, p. 042 001, Feb. 2017. arXiv: 1604.04324 [astro-ph.IM].

[74] S. Sachdev *et al.*, "The GstLAL Search Analysis Methods for Compact Binary Mergers in Advanced LIGO's Second and Advanced Virgo's First Observing Runs," *arXiv e-prints*, arXiv:1901.08580, arXiv:1901.08580, Jan. 2019. arXiv: 1901.08580 [gr-qc].

[75] C. Hanna *et al.*, "Fast evaluation of multidetector consistency for real-time gravitational wave searches," Phys. Rev. D, vol. 101, no. 2, 022003, p. 022 003, Jan. 2020. arXiv: 1901.02227 [gr-qc].

[76] Q. Chu, "Low-latency detection and localization of gravitational waves from compact binary coalescences," Ph.D. dissertation, The University of Western Australia, 2017.

[77] T. Adams *et al.*, "Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era," *Classical and Quantum Gravity*, vol. 33, no. 17, 175012, p. 175 012, Sep. 2016. arXiv: 1512.02864 [gr-qc].

[78] A. H. Nitz, T. Dent, T. Dal Canton, S. Fairhurst, and D. A. Brown, "Detecting binary compact-object mergers with gravitational waves: Understanding and improving the sensitivity of the pycbc search," *The Astrophysical Journal*, vol. 849, no. 2, p. 118, 2017.

[79] A. C. Searle, "Monte-Carlo and Bayesian techniques in gravitational wave burst data analysis," *arXiv e-prints*, arXiv:0804.1161, arXiv:0804.1161, Apr. 2008. arXiv: 0804.1161 [gr-qc].

[80] R. Biswas *et al.*, "Likelihood-ratio ranking of gravitational-wave candidates in a non-Gaussian background," Phys. Rev. D, vol. 85, no. 12, 122008, p. 122 008, Jun. 2012. arXiv: 1201.2959 [gr-qc].

[81] T. Dent and J. Veitch, "Optimizing gravitational-wave searches for a population of coalescing binaries: Intrinsic parameters," Phys. Rev. D, vol. 89, no. 6, 062002, p. 062 002, Mar. 2014. arXiv: 1311.7174 [gr-qc].

[82] B. J. Owen, "Search templates for gravitational waves from inspiraling binaries: Choice of template spacing," Phys. Rev. D, vol. 53, no. 12, pp. 6749–6761, Jun. 1996. arXiv: gr-qc/9511032 [gr-qc].

[83] B. S. Sathyaprakash and S. V. Dhurandhar, "Choice of filters for the detection of gravitational waves from coalescing binaries," Phys. Rev. D, vol. 44, no. 12, pp. 3819–3834, Dec. 1991.

[84] S. V. Dhurandhar and B. S. Sathyaprakash, "Choice of filters for the detection of gravitational waves from coalescing binaries. II. Detection in colored noise," Phys. Rev. D, vol. 49, no. 4, pp. 1707–1722, Feb. 1994.

[85] B. P. Abbott *et al.*, "Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914," *Classical and Quantum Gravity*, vol. 33, no. 13, 134001, p. 134 001, Jul. 2016. arXiv: 1602.03844 [gr-qc].

[86] B. P. Abbott *et al.*, "A guide to LIGO-Virgo detector noise and extraction of transient gravitational-wave signals," *Classical and Quantum Gravity*, vol. 37, no. 5, 055002, p. 055 002, Mar. 2020. arXiv: 1908.11170 [gr-qc].

[87] V. Gayathri *et al.*, "GW190521 as a Highly Eccentric Black Hole Merger," *arXiv e-prints*, arXiv:2009.05461, arXiv:2009.05461, Sep. 2020. arXiv: 2009.05461 [astro-ph.HE].

[88] T. Gebhard, N. Kilbertus, G. Parascandolo, I. Harry, and B. Schölkopf, "Convwave: Searching for gravitational waves with fully convolutional neural nets," in *Workshop on Deep Learning for Physical Sciences (DLPS) at the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 1–6.

[89] D. George and E. Huerta, "Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data," *Physics Letters B*, vol. 778, pp. 64–70, 2018.

[90] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, "Matching matched filtering with deep networks for gravitational-wave astronomy," *Physical review letters*, vol. 120, no. 14, p. 141 103, 2018.

[91] D. George and E. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Physical Review D*, vol. 97, no. 4, p. 044 039, 2018.

[92] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, "Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors," *SCIENCE CHINA Physics, Mechanics & Astronomy*, vol. 62, no. 6, p. 969 512, 2019.

[93] F. Morawski, M. Bejger, and P. Ciecielag, "Convolutional neural network classifier for the output of the time-domain-statistic all-sky search for continuous gravitational waves," *Machine Learning: Science and Technology*, vol. 1, no. 2, p. 025 016, 2020.

[94] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, "Deep-learning continuous gravitational waves," *Physical Review D*, vol. 100, no. 4, p. 044 009, 2019.

[95] P. G. Krastev, "Real-time detection of gravitational waves from binary neutron stars using artificial neural networks," *Physics Letters B*, p. 135 330, 2020.

[96] B.-J. Lin, X.-R. Li, and W.-L. Yu, "Binary neutron stars gravitational wave detection based on wavelet packet analysis and convolutional neural networks," *Frontiers of Physics*, vol. 15, no. 2, pp. 1–7, 2020.

[97] Y.-C. Lin and J.-H. P. Wu, "Detection of gravitational waves using bayesian neural networks," *arXiv preprint arXiv:2007.04176*, 2020.

[98] C. Bresten and J.-H. Jung, "Detection of gravitational waves using topological data analysis and convolutional neural network: An improved approach," *arXiv preprint arXiv:1910.08245*, 2019.

[99] P Astone *et al.*, "New method to observe gravitational waves emitted by core collapse supernovae," *Physical Review D*, vol. 98, no. 12, p. 122 002, 2018.

[100] T. S. Yamamoto and T. Tanaka, "Use of excess power method and convolutional neural network in all-sky search for continuous gravitational waves," *arXiv preprint arXiv:2011.12522*, 2020.

[101] C. Dreissigacker and R. Prix, "Deep-learning continuous gravitational waves: Multiple detectors and realistic noise," *Physical Review D*, vol. 102, no. 2, p. 022 005, 2020.

[102] R. Corizzo, M. Ceci, E. Zdravevski, and N. Japkowicz, "Scalable auto-encoders for gravitational waves detection from time series data," *Expert Systems with Applications*, vol. 151, p. 113 378, 2020.

[103] A. L. Miller *et al.*, "How effective is machine learning to detect long transient gravitational waves from neutron stars in a real search?" *Physical Review D*, vol. 100, no. 6, p. 062 005, 2019.

[104] J. Bayley, C. Messenger, and G. Woan, "Robust machine learning algorithm to search for continuous gravitational waves," *Physical Review D*, vol. 102, no. 8, p. 083 024, 2020.

[105] P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, "Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real ligo data using deep learning," *arXiv preprint arXiv:2012.13101*, 2020.

[106] H.-M. Luo, W. Lin, Z.-C. Chen, and Q.-G. Huang, "Extraction of gravitational wave signals with optimized convolutional neural network," *Frontiers of Physics*, vol. 15, no. 1, pp. 1–6, 2020.

[107] G. R. Santos, M. P. Figueiredo, A. d. P. Santos, P. Protopapas, and T. A. Ferreira, "Gravitational wave detection and information extraction via neural networks," *arXiv preprint arXiv:2003.09995*, 2020.

[108] M. L. Chan, I. S. Heng, and C. Messenger, "Detection and classification of supernova gravitational wave signals: A deep learning approach," *Physical Review D*, vol. 102, no. 4, p. 043 022, 2020.

[109] H. Xia, L. Shao, J. Zhao, and Z. Cao, "Improved deep learning techniques in gravitational-wave data analysis," *Physical Review D*, vol. 103, no. 2, p. 024 040, 2021.

[110] R. Biswas *et al.*, "Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data," *Physical Review D*, vol. 88, no. 6, p. 062 003, 2013.

[111] D. George, H. Shen, and E. Huerta, "Deep transfer learning: A new deep learning glitch classification method for advanced ligo," *arXiv preprint arXiv:1706.07446*, 2017.

[112] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, "Transient classification in ligo data using difference boosting neural network," *Physical Review D*, vol. 95, no. 10, p. 104 059, 2017.

[113] M. Razzano and E. Cuoco, "Image-based deep learning for classification of noise transients in gravitational wave detectors," *Classical and Quantum Gravity*, vol. 35, no. 9, p. 095 016, 2018.

[114] S Coughlin *et al.*, "Classifying the unknown: Discovering novel gravitational-wave detector glitches using similarity learning," *Physical Review D*, vol. 99, no. 8, p. 082 002, 2019.

[115] R. E. Colgan *et al.*, "Efficient gravitational-wave glitch identification from environmental data through machine learning," *Physical Review D*, vol. 101, no. 10, p. 102 003, 2020.

[116] H. Nakano *et al.*, "Comparison of various methods to extract ringdown frequency from gravitational wave data," *Physical Review D*, vol. 99, no. 12, p. 124 032, 2019.

[117] S. R. Green, C. Simpson, and J. Gair, "Gravitational-wave parameter estimation with autoregressive neural network flows," *Physical Review D*, vol. 102, no. 10, p. 104 057, 2020.

[118] J. P. Marulanda, C. Santa, and A. E. Romano, "Deep learning merger masses estimation from gravitational waves signals in the frequency domain," *Physics Letters B*, vol. 810, p. 135 790, 2020.

[119] A Caramete *et al.*, "Characterization of gravitational waves signals using neural networks," *arXiv preprint arXiv:2009.06109*, 2020.

[120] A. Delaunoy *et al.*, "Lightning-fast gravitational wave parameter inference through neural amortization," *arXiv preprint arXiv:2010.12931*, 2020.

[121] H. Shen, D. George, E. A. Huerta, and Z. Zhao, "Denoising gravitational waves with enhanced deep recurrent denoising auto-encoders," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 3237–3241.

[122] W. Wei and E. Huerta, "Gravitational wave denoising of binary black hole mergers with deep learning," *Physics Letters B*, vol. 800, p. 135 081, 2020.

[123] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, "Convolutional neural networks: A magic bullet for gravitational-wave detection?" *Physical Review D*, vol. 100, no. 6, p. 063 015, 2019.

[124] V Gayathri *et al.*, "Eccentricity estimate for black hole mergers with numerical relativity simulations," *Nature Astronomy*, pp. 1–6, 2022.

[125] J. Yan *et al.*, "Generalized approach to matched filtering using neural networks," *arXiv preprint arXiv:2104.03961*, 2021.

[126] J. Yan, R. Colgan, J. Wright, Z. Márka, I. Bartos, and S. Márka, "Boosting the efficiency of parametric detection with hierarchical neural networks," *Physical Review D*, vol. 106, no. 6, p. 063 008, 2022.

[127] R. Balestriero and R. Baraniuk, "Mad max: Affine spline insights into deep learning," *arXiv preprint arXiv:1805.06576*, 2018.

[128] J. Cheng, Y. Wu, W. AbdAlmageed, and P. Natarajan, "Qatm: Quality-aware template matching for deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 553–11 562.

[129] J. M. Bower, Y.-F. Wong, and J. Banik, "Neural networks for template matching: Application to real-time classification of the action potentials of real neurons," in *Neural Information Processing Systems*, 1988, pp. 103–113.

[130] D. Tank and J. Hopfield, "Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 533–541, 1986.

[131] D. Buniatyan, T. Macrina, D. Ih, J. Zung, and H. S. Seung, "Deep learning improves template matching by normalized cross correlation," *arXiv preprint arXiv:1705.08593*, 2017.

[132] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-network-based adaptive matched filtering for qrs detection," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 4, pp. 317–329, 1992.

[133] R. P. Lippmann and P. Beckman, "Adaptive neural net preprocessing for signal detection in non-gaussian noise," in *Advances in neural information processing systems*, 1989, pp. 124–132.

[134] R. A. et al, "Open data from the first and second observing runs of advanced ligo and advanced virgo," *SoftwareX*, vol. 13, p. 100 658, 2021.

[135] Q. Yu, "A general method of finding a minimax estimator of a distribution function when no equalizer rule is available," *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, pp. 281–290, 1992.

[136] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[137] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *International conference on artificial neural networks*, Springer, 2010, pp. 92–101.

[138] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Massachusetts, USA: 2017, vol. 1.

[139] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[140] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural networks*, vol. 6, no. 6, pp. 861–867, 1993.

[141] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," *Advances in neural information processing systems*, vol. 30, pp. 6231–6239, 2017.

[142] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[143] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 370–378.

[144] J. Sun, H. Li, Z. Xu, *et al.*, "Deep admm-net for compressive sensing mri," in *Advances in neural information processing systems*, 2016, pp. 10–18.

[145] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Conference on Learning Theory*, PMLR, 2020, pp. 2306–2327.

[146] P. Orponen *et al.*, "Computational complexity of neural networks: A survey," *Nordic Journal of Computing*, 1994.

[147] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 8, pp. 1553–1565, 2014.

[148] J. Creighton and W. Anderson, *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis.* 2011.

[149] R. Balasubramanian, B. S. Sathyaprakash, and S. V. Dhurandhar, "Gravitational waves from coalescing binaries: Detection strategies and Monte Carlo estimation of parameters," Phys. Rev. D, vol. 53, no. 6, pp. 3033–3055, Mar. 1996. arXiv: gr-qc/9508011 [gr-qc].

[150] P. R. Brady, T. Creighton, C. Cutler, and B. F. Schutz, "Searching for periodic sources with LIGO," Phys. Rev. D, vol. 57, no. 4, pp. 2101–2116, Feb. 1998. arXiv: gr-qc/9702050 [gr-qc].

[151] C. Messenger, R. Prix, and M. A. Papa, "Random template banks and relaxed lattice coverings," Phys. Rev. D, vol. 79, no. 10, 104017, p. 104 017, May 2009. arXiv: 0809.5223 [gr-qc].

[152] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[153] S. Mohanty and S. Dhurandhar, "Hierarchical search strategy for the detection of gravitational waves from coalescing binaries," *Physical Review D*, vol. 54, no. 12, p. 7108, 1996.

[154] A. S. Sengupta, S. Dhurandhar, and A. Lazzarini, "Faster implementation of the hierarchical search algorithm for detection of gravitational waves from inspiraling compact binaries," *Physical Review D*, vol. 67, no. 8, p. 082 004, 2003.

[155] A. S. Sengupta, S. V. Dhurandhar, A. Lazzarini, and T. Prince, "Extended hierarchical search (ehs) algorithm for detection of gravitational waves from inspiralling compact binaries," *Classical and Quantum Gravity*, vol. 19, no. 7, p. 1507, 2002.

[156] B. Gadre, S. Mitra, and S. Dhurandhar, "Hierarchical search strategy for the efficient detection of gravitational waves from nonprecessing coalescing compact binaries with aligned-spins," *Physical Review D*, vol. 99, no. 12, p. 124 035, 2019.

[157] R. Dhurkunde, H. Fehrmann, and A. H. Nitz, "A hierarchical approach to matched filtering using a reduced basis," *arXiv preprint arXiv:2110.13115*, 2021.

[158] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, IEEE, vol. 2, 2005, pp. 1458–1465.

[159] Q. Chen, Z. Song, Y. Hua, Z. Huang, and S. Yan, "Hierarchical matching with side information for image classification," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 3426–3433.

[160] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 2169–2178.

[161] C. Shen *et al.*, "Sentiment classification towards question-answering with hierarchical matching network," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3654–3663.

[162] M. Vaillant and C. Davatzikos, "Hierarchical matching of cortical features for deformable brain image registration," in *Biennial International Conference on Information Processing in Medical Imaging*, Springer, 1999, pp. 182–195.

[163] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon, "Hierarchical part-template matching for human detection and segmentation," in *2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[164] P. Wang, Y. Fan, S. Niu, Z. Yang, Y. Zhang, and J. Guo, "Hierarchical matching network for crime classification," in *proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 325–334.

[165] T. Dent and J. Veitch, "Optimizing gravitational-wave searches for a population of coalescing binaries: Intrinsic parameters," *Physical Review D*, vol. 89, no. 6, p. 062 002, 2014.

[166] *Gwosc summary*, `https://ldas-jobs.ligo-la.caltech.edu/~detchar/summary/day/20170801/`.

[167] M. Cavaglia, K. Staats, and T. Gill, "Finding the origin of noise transients in ligo data with machine learning," *arXiv preprint arXiv:1812.05225*, 2018.

[168] R. E. Colgan *et al.*, "Efficient gravitational-wave glitch identification from environmental data through machine learning," Phys. Rev. D, vol. 101, no. 10, 102003, p. 102 003, May 2020. arXiv: `1911.11831 [astro-ph.IM]`.

[169] R. E. Colgan, J. Yan, Z. Márka, I. Bartos, S. Márka, and J. N. Wright, "Architectural Optimization and Feature Learning for High-Dimensional Time Series Datasets," *arXiv e-prints*, arXiv:2202.13486, arXiv:2202.13486, Feb. 2022. arXiv: `2202.13486 [cs.LG]`.

[170] R. E. Colgan, Z. Márka, J. Yan, I. Bartos, J. N. Wright, and S. Márka, "Detecting and Diagnosing Terrestrial Gravitational-Wave Mimics Through Feature Learning," *arXiv e-prints*, arXiv:2203.05086, arXiv:2203.05086, Mar. 2022. arXiv: `2203.05086 [astro-ph.IM]`.

[171] E. Cuoco *et al.*, "Enhancing gravitational-wave science with machine learning," *Machine Learning: Science and Technology*, vol. 2, no. 1, p. 011 002, Dec. 2020.

[172] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 1791–1798.

[173] E. A. Gibson *et al.*, "Principal component pursuit for pattern identification in environmental mixtures," *Environmental Health Perspectives*, vol. 130, no. 11, p. 117 008, 2022.

[174] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, AAAI Press/International Joint Conferences on Artificial Intelligence, 2015, pp. 2270–2276.

[175] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transforms," *Pattern Recognition*, vol. 35, no. 1, pp. 31–41, 2002.

[176] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: State-of-the-art and future challenges," *Nature neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.

[177] D. Lunga, S. Prasad, M. M. Crawford, and O. Ersoy, "Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 55–66, 2013.

[178] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk, "The multiscale structure of non-differentiable image manifolds," in *Wavelets XI*, SPIE, vol. 5914, 2005, pp. 413–429.

[179] M. B. Wakin, "The geometry of low-dimensional signal models," Ph.D. dissertation, Rice University, 2007.

[180] R. G. Baraniuk and M. B. Wakin, "Random projections of smooth manifolds," *Foundations of computational mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[181] T Akutsu *et al.*, "Overview of kagra: Detector design and construction history," *Progress of Theoretical and Experimental Physics*, vol. 2021, no. 5, 05A101, 2021.

[182] B. P. Abbott *et al.*, "Observation of gravitational waves from a binary black hole merger," *Physical review letters*, vol. 116, no. 6, p. 061 102, 2016.

[183] B. P. Abbott *et al.*, "Gw170817: Observation of gravitational waves from a binary neutron star inspiral," *Physical review letters*, vol. 119, no. 16, p. 161 101, 2017.

[184] Y. Bengio and M. Monperrus, "Non-local manifold tangent learning," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[185] B. Culpepper and B. Olshausen, "Learning transport operators for image manifolds," *Advances in neural information processing systems*, vol. 22, 2009.

[186] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th international conference on international conference on machine learning*, 2011, pp. 833–840.

[187] M. Park, W. Jitkrittum, A. Qamar, Z. Szabó, L. Buesing, and M. Sahani, "Bayesian manifold learning: The locally linear latent variable model (ll-lvm)," *Advances in neural information processing systems*, vol. 28, 2015.

[188] A. Kumar, P. Sattigeri, and T. Fletcher, "Semi-supervised learning with gans: Manifold invariance with improved inference," *Advances in neural information processing systems*, vol. 30, 2017.

[189] T. Chen *et al.*, "Learning to optimize: A primer and a benchmark," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 8562–8620, 2022.

[190] J. Liu and X. Chen, "Alista: Analytic weights are as good as learned weights in lista," in *International Conference on Learning Representations (ICLR)*, 2019.

[191] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?" *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[192] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[193] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," *arXiv preprint arXiv:1705.08041*, 2017.

[194] D. Liang, J. Cheng, Z. Ke, and L. Ying, "Deep mri reconstruction: Unrolled optimization algorithms meet neural networks," *arXiv preprint arXiv:1907.11711*, 2019.

[195] S. Buchanan, J. Yan, E. Haber, and J. Wright, "Resource-efficient invariant networks: Exponential gains by unrolled optimization," *arXiv preprint arXiv:2203.05006*, 2022.

[196] C. W. Helstrom, *Statistical theory of signal detection: international series of monographs in electronics and instrumentation*. Elsevier, 2013, vol. 9.

[197] J. Yan *et al.*, "Generalized approach to matched filtering using neural networks," *Physical Review D*, vol. 105, no. 4, p. 043 006, 2022.

[198] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[199] V. A. Toponogov, *Differential geometry of curves and surfaces*. Springer, 2006.

[200] J. M. Lee, *Introduction to Riemannian manifolds*. Springer, 2018, vol. 2.

[201] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[202] Y. Zhang, H.-W. Kuo, and J. Wright, "Structured local optima in sparse blind deconvolution," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 419–452, 2019.

[203] G. D. O'Neil, H.-w. Kuo, D. N. Lomax, J. Wright, and D. V. Esposito, "Scanning line probe microscopy: Beyond the point probe," *Analytical chemistry*, vol. 90, no. 19, pp. 11 531–11 537, 2018.

[204] A. E. Dorfi, J. Yan, J. Wright, and D. V. Esposito, "Compressed sensing image reconstruction of scanning electrochemical microscopy measurements carried out at ultrahigh scan speeds using continuous line probes," *Analytical Chemistry*, vol. 93, no. 37, pp. 12 574–12 581, 2021.

[205] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[206] R van Handel, "Apc 550: Probability in high dimension," *Lecture Notes. Princeton University. Retrieved from https://web. math. princeton. edu/rvan/APC550. pdf on December*, vol. 21, p. 2016, 2016.

[207] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

[208] J. M. Lee, *Riemannian Manifolds: an Introduction to Curvature*. Springer, 1997.

[209] A. Nitz *et al.*, *Gwastro/pycbc: V2.1.2 release of pycbc*, version v2.1.2, May 2023.

Appendix A: Proofs for Generalized Approach to Matched Filtering Using Neural Networks

A.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

$$\lambda(\boldsymbol{x}) = \frac{\int \rho_{\text{noise}}(\boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{\gamma}})\, \mathrm{d}\nu(\boldsymbol{\gamma})}{\rho_{\text{noise}}(\boldsymbol{x})} \tag{A.1}$$

$$= \int \frac{\rho_{\text{noise}}(\boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{\gamma}})}{\rho_{\text{noise}}(\boldsymbol{x})}\, \mathrm{d}\nu(\boldsymbol{\gamma}). \tag{A.2}$$

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

$$\frac{\rho_{\text{noise}}(\boldsymbol{x} - \boldsymbol{s}_{\boldsymbol{\gamma}})}{\rho_{\text{noise}}(\boldsymbol{x})} = \exp\left(\frac{\left\langle \boldsymbol{x}, \boldsymbol{s}_{\boldsymbol{\gamma}} \right\rangle - \|\boldsymbol{s}_{\boldsymbol{\gamma}}\|^2/2}{\sigma^2}\right), \tag{A.3}$$

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.

A.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

$$\begin{aligned}
\mathcal{R}(f) &= \mathbb{E}_{(\boldsymbol{x},y)}[\ell(f(\boldsymbol{x}), y)] && \text{(A.4)}\\
&= \mathbb{P}[y = 0] \cdot \mathbb{E}_{\boldsymbol{x}|y=0}[\ell(f(\boldsymbol{x}), 0)] + \\
&\quad \mathbb{P}[y = 1] \cdot \mathbb{E}_{\boldsymbol{x}|y=1}[\ell(f(\boldsymbol{x}), 1)] && \text{(A.5)}\\
&= \mathbb{P}[y = 0] \int_{\mathbb{R}^n} \ell(f(\boldsymbol{x}), 0) p_0(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \\
&\quad \mathbb{P}[y = 1] \int_{\mathbb{R}^n} \ell(f(\boldsymbol{x}), 1) p_1(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} && \text{(A.6)}\\
&= \int_{\mathbb{R}^n} \Big((1 - c)\ell(f(\boldsymbol{x}), 0) p_0(\boldsymbol{x}) + \\
&\quad c\ell(f(\boldsymbol{x}), 1) p_1(\boldsymbol{x}) \Big) \mathrm{d}\boldsymbol{x}, && \text{(A.7)}
\end{aligned}$$

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

$$f_\star(\boldsymbol{x}) = \arg\min_{\hat{y}} (1 - c)\ell(\hat{y}, 0) p_0(\boldsymbol{x}) + c\ell(\hat{y}, 1) p_1(\boldsymbol{x}) \tag{A.8}$$

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

$$f_\star(\boldsymbol{x}) = \arg\min_{\hat{y}} \ \ell(\hat{y}, 0) + \frac{c\lambda(\boldsymbol{x})}{1 - c} \ell(\hat{y}, 1). \tag{A.9}$$

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 equation (A.9) 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

$$\frac{\ell'(\hat{y}, 0)}{-\ell'(\hat{y}, 1)} = \frac{c\lambda(\boldsymbol{x})}{1-c}. \tag{A.10}$$

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 (A.10) is strictly increasing in $\hat{y}$.

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

Appendix B: Proofs for TpopT: Efficient Trainable Template Optimization on Low-Dimensional Manifolds

B.1 Overview

In the appendices, we will prove Theorem 3 from the main paper.

For the rest of the supplementary materials, Section B.2 proves result (4.10) under the stricter constraint on the noise level σ, Section B.3 bounds the effect of noise on the tangent bundle, and Section B.4 uses this bound to prove result (4.9) under the looser constraint on the noise level.

B.2 Proof of Result (4.10)

In this section, we state and prove one of the two parts of our main claims about gradient descent:

Theorem 4. *Let S be a complete manifold. Suppose the extrinsic geodesic curvature of S is bounded by κ. Consider the Riemannian gradient method, with initialization satisfying $d(\boldsymbol{s}^0, \boldsymbol{s}_\natural) < \Delta = 1/\kappa$, and step size $\tau = \frac{1}{64}$. Then when $\sigma \leq 1/(60\kappa\sqrt{D})$, with probability at least $1 - e^{-D/2}$, we have for all k*

$$d(\boldsymbol{s}^k, \boldsymbol{s}^\star) \leq \big(1 - \epsilon\big)^k d(\boldsymbol{s}^0, \boldsymbol{s}^\star), \tag{B.1}$$

where $\boldsymbol{s}^\star$ is the unique minimizer of f over $B(\boldsymbol{s}_\natural, 1/\kappa)$. Here, c, ϵ are positive numerical constants.

Proof. Since the closed neighborhood $B(\boldsymbol{s}_\natural, 1/\kappa)$ is a compact set and f is continuous, there must exist a minimizer of f on $B(\boldsymbol{s}_\natural, 1/\kappa)$, which we denote as $\boldsymbol{s}^\star$. We will show that with high proba-

bility $\boldsymbol{s}^\star$ does not lie on the boundary $\partial B(\boldsymbol{s}_\natural, 1/\kappa)$. It suffices to show that $\forall \boldsymbol{s} \in \partial B(\boldsymbol{s}_\natural, 1/\kappa)$:

$$\left\langle -\operatorname{grad}[f](\boldsymbol{s}),\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle > 0, \tag{B.2}$$

namely that the gradient descent direction points inward the neighborhood for all points on the boundary. Here $\log_{\boldsymbol{s}} : S \to T_{\boldsymbol{s}}S$ denotes the logarithmic map at point $\boldsymbol{s} \in S$. To show this, we have

$$\begin{aligned}
\left\langle -\operatorname{grad}[f](\boldsymbol{s}),\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle &= \left\langle P_{T_{\boldsymbol{s}}S}[\boldsymbol{s}_\natural + \boldsymbol{z}],\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle \\
&= \left\langle \boldsymbol{s}_\natural + \boldsymbol{z},\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle \\
&\geq \left\langle \boldsymbol{s}_\natural,\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle - \|\boldsymbol{z}\|_2,
\end{aligned} \tag{B.3}$$

where the operator $P_{T_{\boldsymbol{s}}S}[\cdot]$ denotes projection onto the tangent space at $\boldsymbol{s}$, and we used the fact that $\log_{\boldsymbol{s}} \boldsymbol{s}_\natural \in T_{\boldsymbol{s}}S$. Let $\boldsymbol{\gamma}$ be a unit-speed geodesic of S with $\boldsymbol{\gamma}(0) = \boldsymbol{s}_\natural$ and $\boldsymbol{\gamma}(\Delta) = \boldsymbol{s}$, the existence of which is ensured by the completeness of S. Hence $\dot{\boldsymbol{\gamma}}(\Delta) = -\frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2}$. Using Lemma 5, it follows that

$$\left\langle \boldsymbol{s}_\natural,\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle = \langle \boldsymbol{\gamma}(0), -\dot{\boldsymbol{\gamma}}(\Delta) \rangle \geq \Delta - \frac{1}{6}\kappa^2\Delta^3 = \frac{5}{6\kappa}. \tag{B.4}$$

Throughout this proof, we will use the result from measure concentration that $\|\boldsymbol{z}\|_2 \leq 2\sigma\sqrt{D}$ with probability at least $1-e^{-D/2}$ [205]. Hence with high probability we have $\left\langle -\operatorname{grad}[f](\boldsymbol{s}),\ \frac{\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle \geq \frac{5}{6\kappa} - \frac{1}{30\kappa} > 0$. Therefore, with high probability $\boldsymbol{s}^\star$ lies in the interior of $B(\boldsymbol{s}_\natural, 1/\kappa)$, and hence the gradient vanishes at $\boldsymbol{s}^\star$, i.e. $\operatorname{grad}[f](\boldsymbol{s}^\star) = \boldsymbol{0}$.

Suppose we are currently at the k-th iteration with iterate $\boldsymbol{s}^k$. Define $\boldsymbol{s}_t = \exp_{\boldsymbol{s}^k}\left(-t \operatorname{grad}[f](\boldsymbol{s}^k)\right)$ with variable $t \in [0, \tau]$, and the next iterate can be represented as $\boldsymbol{s}^{k+1} = \boldsymbol{s}_\tau$. The global definition

of the exponential map is ensured by the completeness of S. We have that

$$\begin{aligned}
d(\boldsymbol{s}^{k+1}, \boldsymbol{s}^\star) - d(\boldsymbol{s}^k, \boldsymbol{s}^\star) &= \int_0^\tau \frac{d}{dr} d(\boldsymbol{s}_r, \boldsymbol{s}^\star)\Big|_t \, dt \\
&= \int_0^\tau \left\langle \frac{d}{dr}\boldsymbol{s}_r\Big|_t, \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&= \int_0^\tau \left\langle \Pi_{\boldsymbol{s}_t, \boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\}, \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&= \int_0^\tau \left\langle -\operatorname{grad}[f](\boldsymbol{s}_t), \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&\quad + \int_0^\tau \left\langle \operatorname{grad}[f](\boldsymbol{s}_t) + \Pi_{\boldsymbol{s}_t, \boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\}, \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&= \int_0^\tau \left\langle P_{T_{\boldsymbol{s}_t} S}[\boldsymbol{s}^\star], \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&\quad + \int_0^\tau \left\langle P_{T_{\boldsymbol{s}_t} S}[\boldsymbol{x} - \boldsymbol{s}^\star], \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt \\
&\quad + \int_0^\tau \left\langle \operatorname{grad}[f](\boldsymbol{s}_t) + \Pi_{\boldsymbol{s}_t, \boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\}, \frac{-\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t} \boldsymbol{s}^\star\|_2} \right\rangle dt.
\end{aligned} \tag{B.5}$$

The third equation holds because the velocity at the new point $\boldsymbol{s}_t$ is the same velocity vector at $\boldsymbol{s}^k$ but transported along the curve since there is no acceleration along the curve. The last equation follows from the fact that $\operatorname{grad}[f](\boldsymbol{s}_t) = -P_{T_{\boldsymbol{s}_t} S}[\boldsymbol{x}]$. In the following, we will bound the three terms in (B.5) separately.

For convenience, write

$$d(t) = d(\boldsymbol{s}_t, \boldsymbol{s}^\star) \tag{B.6}$$

so that $d(0) = d(\boldsymbol{s}^k, \boldsymbol{s}^\star)$ and $d(\tau) = d(\boldsymbol{s}^{k+1}, \boldsymbol{s}^\star)$.

For the integrand of the first term in (B.5), let $\boldsymbol{\gamma}$ be a unit-speed geodesic between $\boldsymbol{s}_t$ and $\boldsymbol{s}^\star$,

where $\boldsymbol{\gamma}(0) = \boldsymbol{s}^\star$ and $\boldsymbol{\gamma}(d(t)) = \boldsymbol{s}_t$. We have

$$\begin{aligned}\left\langle P_{T_{\boldsymbol{s}_t}S}[\boldsymbol{s}^\star],\ \frac{-\log_{\boldsymbol{s}_t}\boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t}\boldsymbol{s}^\star\|_2}\right\rangle &= \left\langle \boldsymbol{s}^\star,\ \frac{-\log_{\boldsymbol{s}_t}(\boldsymbol{s}^\star)}{\|\log_{\boldsymbol{s}_t}(\boldsymbol{s}^\star)\|_2}\right\rangle \\ &= \langle \boldsymbol{\gamma}(0), \dot{\boldsymbol{\gamma}}(d(t))\rangle \\ &\le -d(t) + \frac{1}{6}\kappa^2 d^3(t), \end{aligned} \tag{B.7}$$

where we used the fact that $\dot{\boldsymbol{\gamma}}(d(t)) = \frac{-\log_{\boldsymbol{s}_t}(\boldsymbol{s}^\star)}{\|\log_{\boldsymbol{s}_t}(\boldsymbol{s}^\star)\|_2} \in T_{\boldsymbol{s}_t}S$, and the inequality is given by Lemma 5.

For the integrand of the second term in (B.5), we have

$$\begin{aligned}\left\langle P_{T_{\boldsymbol{s}_t}S}[\boldsymbol{x}-\boldsymbol{s}^\star],\ \frac{-\log_{\boldsymbol{s}_t}\boldsymbol{s}^\star}{\|\log_{\boldsymbol{s}_t}\boldsymbol{s}^\star\|_2}\right\rangle &\le \|P_{T_{\boldsymbol{s}_t}S}[\boldsymbol{x}-\boldsymbol{s}^\star]\|_2 \\ &= \|P_{T_{\boldsymbol{s}_t}S}\ P_{(T_{\boldsymbol{s}^\star}S)^\perp}[\boldsymbol{x}-\boldsymbol{s}^\star]\|_2 \\ &\le \|P_{T_{\boldsymbol{s}_t}S}\ P_{(T_{\boldsymbol{s}^\star}S)^\perp}\|\ \|\boldsymbol{x}-\boldsymbol{s}^\star\|_2 \\ &\le \|P_{(T_{\boldsymbol{s}^\star}S)^\perp}\ P_{T_{\boldsymbol{s}_t}S}\|\ \|\boldsymbol{z}\|_2, \end{aligned} \tag{B.8}$$

where we used the optimality of $\boldsymbol{s}^\star$ and the symmetry of projection operators. The operator norm $\|P_{(T_{\boldsymbol{s}^\star}S)^\perp}\ P_{T_{\boldsymbol{s}_t}S}\|$ can be rewritten as

$$\|P_{(T_{\boldsymbol{s}^\star}S)^\perp}\ P_{T_{\boldsymbol{s}_t}S}\| = \sup_{\boldsymbol{v}\in T_{\boldsymbol{s}_t}S,\ \|\boldsymbol{v}\|_2=1} d(\boldsymbol{v}, T_{\boldsymbol{s}^\star}S). \tag{B.9}$$

For any unit vector $\boldsymbol{v} \in T_{\boldsymbol{s}_t}S$, we will construct a vector in $T_{\boldsymbol{s}^\star}S$ and use its distance from $\boldsymbol{v}$ to upper bound $d(\boldsymbol{v}, T_{\boldsymbol{s}^\star}S)$. We again use the unit-speed geodesic $\boldsymbol{\gamma}$ joining $\boldsymbol{s}^\star$ and $\boldsymbol{s}_t$, where $\boldsymbol{\gamma}(0) = \boldsymbol{s}^\star$ and $\boldsymbol{\gamma}(d(t)) = \boldsymbol{s}_t$. Let $\boldsymbol{v}_r = \mathcal{P}_{r,d(t)}\boldsymbol{v}$ for $r \in [0, d(t)]$, where $\mathcal{P}_{r,d(t)}$ denotes the parallel transport backward along $\boldsymbol{\gamma}$. The derivative of $\boldsymbol{v}_r$ can be expressed by the second fundamental form $\frac{d}{dr}\boldsymbol{v}_r = \mathrm{I\!I}(\dot{\boldsymbol{\gamma}}(r),\ \boldsymbol{v}_r)$, which can further be bounded by Lemma 6 to get $\|\frac{d}{dr}\boldsymbol{v}_r\|_2 \le 3\kappa$. Hence $\left\|\boldsymbol{v}_{d(t)} - \boldsymbol{v}_0\right\|_2 \le 3\kappa d(t)$. Since $\boldsymbol{v}_{d(t)} = \boldsymbol{v}$ and $\boldsymbol{v}_0 \in T_{\boldsymbol{s}^\star}S$, it follows that $d(\boldsymbol{v}, T_{\boldsymbol{s}^\star}S) \le 3\kappa d(t)$ for any unit vector $\boldsymbol{v} \in T_{\boldsymbol{s}_t}S$. Hence

$$\|P_{(T_{\boldsymbol{s}^\star}S)^\perp}\ P_{T_{\boldsymbol{s}_t}S}\| \le 3\kappa \cdot d(t). \tag{B.10}$$

Since $\|z\|_2 \leq 2\sigma\sqrt{D}$ with high probability, plugging these into (B.8), we have with high probability

$$\left\langle P_{T_{s_t}S}[\boldsymbol{x} - \boldsymbol{s}^\star],\ \frac{-\log_{s_t} \boldsymbol{s}^\star}{\|\log_{s_t} \boldsymbol{s}^\star\|_2} \right\rangle \leq 6\sigma\kappa\sqrt{D} \cdot d(t). \tag{B.11}$$

The integrand of the third term in (B.5) can be bounded using the Riemannian Hessian. We have

$$\operatorname{grad}[f](\boldsymbol{s}_t) = \Pi_{\boldsymbol{s}_t,\boldsymbol{s}^k}\{\operatorname{grad}[f](\boldsymbol{s}^k)\} + \int_{r=0}^{t} \Pi_{\boldsymbol{s}_t,\boldsymbol{s}_r} \operatorname{Hess}[f](\boldsymbol{s}_r)\, \Pi_{\boldsymbol{s}_r,\boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\}\, dr. \tag{B.12}$$

Using the L-Lipschitz gradient property of the function f from Lemma 7, we have

$$\begin{aligned}\left\langle \operatorname{grad}[f](\boldsymbol{s}_t) + \Pi_{\boldsymbol{s}_t,\boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\},\ \frac{-\log_{s_r} s^\star}{\|\log_{s_r} s^\star\|_2} \right\rangle &\leq \|\operatorname{grad}[f](\boldsymbol{s}_t) - \Pi_{\boldsymbol{s}_t,\boldsymbol{s}^k}\{\operatorname{grad}[f](\boldsymbol{s}^k)\}\|_2 \\ &\leq t \max_{\bar{s}} \|\operatorname{Hess}[f](\bar{s})\|\, \|\operatorname{grad}[f](\boldsymbol{s}^k)\|_2 \\ &\leq tL^2 d(\boldsymbol{s}^k, \boldsymbol{s}^\star). \end{aligned} \tag{B.13}$$

Hence

$$\int_0^\tau \left\langle \operatorname{grad}[f](\boldsymbol{s}_t) + \Pi_{\boldsymbol{s}_t,\boldsymbol{s}^k}\{-\operatorname{grad}[f](\boldsymbol{s}^k)\},\ \frac{-\log_{s_r} \boldsymbol{s}^\star}{\|\log_{s_r} \boldsymbol{s}^\star\|_2} \right\rangle dt\ \leq \frac{1}{2}\tau^2 L^2 d(\boldsymbol{s}^k, \boldsymbol{s}^\star). \tag{B.14}$$

Gathering the separate bounds of the three terms in (B.5), we have

$$\begin{aligned} d(\tau) &\leq d(0) + \int_0^\tau \left(-d(t) + \frac{1}{6}\kappa^2 d^3(t) + 6\sigma\kappa\sqrt{D} d(t)\right) dt + \frac{1}{2}L^2\tau^2 d(0) \\ &= (1 + \frac{1}{2}L^2\tau^2) d(0) + \int_0^\tau \left(-(1 - c_1) d(t) + \frac{1}{6}\kappa^2 d^3(t)\right) dt, \end{aligned} \tag{B.15}$$

where $c_1 = 6\sigma\kappa\sqrt{D} \leq \frac{1}{10}$. By triangle inequality we have

$$d(\boldsymbol{s}^k, \boldsymbol{s}^\star) - d(\boldsymbol{s}^k, \boldsymbol{s}_t) \leq d(\boldsymbol{s}_t, \boldsymbol{s}^\star) \leq d(\boldsymbol{s}^k, \boldsymbol{s}^\star) + d(\boldsymbol{s}^k, \boldsymbol{s}_t). \tag{B.16}$$

Since $\boldsymbol{s}_t = \exp_{\boldsymbol{s}^k}\big(-t\,\mathrm{grad}[f](\boldsymbol{s}^k)\big)$, we have

$$d(\boldsymbol{s}^k, \boldsymbol{s}_t) \leq t\|\,\mathrm{grad}[f](\boldsymbol{s}^k)\|_2 \leq tL \cdot d(\boldsymbol{s}^k, \boldsymbol{s}^*), \tag{B.17}$$

and thus

$$(1 - tL)d(0) \leq d(t) \leq (1 + tL)d(0). \tag{B.18}$$

Hence for the integrand in (B.15), we have

$$\begin{aligned} -(1 - c_1)d(t) + \frac{1}{6}\kappa^2 d^3(t) &\leq -(1 - c_1)d(t) + \frac{1}{6}\kappa^2(1 + tL)^2 d^2(0)d(t) \\ &\leq -(1 - c_1)d(t) + \frac{1}{6}\kappa^2(1 + \tau L)^2(2\Delta)^2 d(t) \\ &= (-1 + c_2)d(t) \\ &\leq (-1 + c_2)(1 - tL)d(0) \end{aligned} \tag{B.19}$$

where $c_2 = c_1 + \frac{2}{3}(1 + \tau L)^2$.

Plugging this back, we get

$$\begin{aligned} d(\tau) &\leq (1 + \frac{1}{2}L^2\tau^2)d(0) + (-1 + c_2)d(0)\int_0^\tau (1 - tL)dt \\ &= \left(1 - (1 - c_2)\tau + \left(\frac{1}{2}L^2 + \frac{1}{2}L(1 - c_2)\right)\tau^2\right)d(0). \end{aligned} \tag{B.20}$$

Substituting in $L = \frac{121}{30}$ from Lemma 7 and $\tau = \frac{1}{64}$, we get

$$d(\boldsymbol{s}^{k+1}, \boldsymbol{s}^\star) \leq (1 - \epsilon)d(\boldsymbol{s}^k, \boldsymbol{s}^\star) \tag{B.21}$$

where $\epsilon \approx 2.3 \times 10^{-4}$, which proves result (4.10). Note that this also implies the uniqueness of the minimizer $\boldsymbol{s}^\star$.

□

B.2.1 Supporting Lemmas

Lemma 5. *Let $\boldsymbol{\gamma}$ be a regular unit-speed curve on the manifold $S \subset \mathbb{S}^{d-1}$ with extrinsic curvature κ. Then,*

$$\langle \dot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(0) \rangle \leq -t + \frac{\kappa^2 t^3}{6} \tag{B.22}$$

Proof. Since $\boldsymbol{\gamma} \subset \mathbb{S}^{d-1}$, by differentiating both sides of $\|\boldsymbol{\gamma}(t)\|_2^2 = 1$ we get $\langle \dot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) \rangle = 0$. Further, since $\boldsymbol{\gamma}$ is unit-speed, we have $\|\dot{\boldsymbol{\gamma}}(t)\|_2^2 = 1$ and by differentiating it $\langle \ddot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t) \rangle = 0$. Therefore,

$$\begin{aligned}
\langle \dot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(0) \rangle &= \left\langle \dot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) - \int_0^t \dot{\boldsymbol{\gamma}}(t_1)\, dt_1 \right\rangle \\
&= -\left\langle \dot{\boldsymbol{\gamma}}(t), \int_0^t \dot{\boldsymbol{\gamma}}(t_1)\, dt_1 \right\rangle \\
&= -\int_{t_1=0}^t \left\langle \dot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t) - \int_{t_2=t_1}^t \ddot{\boldsymbol{\gamma}}(t_2)\, dt_2 \right\rangle dt_1 \\
&= -t + \int_{t_1=0}^t \int_{t_2=t_1}^t \langle \dot{\boldsymbol{\gamma}}(t), \ddot{\boldsymbol{\gamma}}(t_2) \rangle \, dt_2\, dt_1 \\
&= -t + \int_{t_1=0}^t \int_{t_2=t_1}^t \left\langle \dot{\boldsymbol{\gamma}}(t_2) + \int_{t_3=t_2}^t \ddot{\boldsymbol{\gamma}}(t_3)\, dt_3, \ddot{\boldsymbol{\gamma}}(t_2) \right\rangle dt_2\, dt_1 \\
&= -t + \int_{t_1=0}^t \int_{t_2=t_1}^t \int_{t_3=t_2}^t \langle \ddot{\boldsymbol{\gamma}}(t_3), \ddot{\boldsymbol{\gamma}}(t_2) \rangle \, dt_3\, dt_2\, dt_1 \\
&\leq -t + \kappa^2 \int_{t_1=0}^t \int_{t_2=t_1}^t \int_{t_3=t_2}^t dt_3\, dt_2\, dt_1 \\
&= -t + \frac{\kappa^2 t^3}{6}.
\end{aligned} \tag{B.23}$$

□

Lemma 6. *Let $\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})$ denote the second fundamental form at some point $\boldsymbol{s} \in S$, and let κ denote the extrinsic ($\mathbb{R}^D$) geodesic curvature of S. Then*

$$\sup_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} \|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2 \leq 3\kappa. \tag{B.24}$$

Proof. Set

$$\kappa^{\mathrm{I\!I}} = \max_{\|\boldsymbol{u}\|_2=1, \|\boldsymbol{v}\|_2=1} \|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2^2. \tag{B.25}$$

Choose unit vectors $\boldsymbol{u}$, $\boldsymbol{v}$ which realize this maximum value (these must exist, by continuity of $\mathrm{I\!I}$ and compactness of the constraint set). Because $\mathrm{I\!I}$ is bilinear, $\|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2^2 = \|\mathrm{I\!I}(\boldsymbol{u}, -\boldsymbol{v})\|_2^2$, and without loss of generality, we can assume $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \leq 0$.

Since $\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})$ is a symmetric bilinear form, each coordinate of the vector $\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})$ has the form $\mathrm{I\!I}_i(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{v}$ for some symmetric $d \times d$ matrix $\boldsymbol{\Phi}_i$. Now,

$$\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{v} = \tfrac{1}{2}(\boldsymbol{u} + \boldsymbol{v})^T \boldsymbol{\Phi}_i (\boldsymbol{u} + \boldsymbol{v}) - \tfrac{1}{2}\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{u} - \tfrac{1}{2}\boldsymbol{v}^T \boldsymbol{\Phi}_i \boldsymbol{v}, \tag{B.26}$$

so

$$|\tfrac{1}{2}\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{u}| + |\tfrac{1}{2}\boldsymbol{v}^T \boldsymbol{\Phi}_i \boldsymbol{v}| + |\tfrac{1}{2}(\boldsymbol{u} + \boldsymbol{v})^T \boldsymbol{\Phi}_i (\boldsymbol{u} + \boldsymbol{v})| \geq |\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{v}| \tag{B.27}$$

and

$$3|\tfrac{1}{2}\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{u}|^2 + 3|\tfrac{1}{2}\boldsymbol{v}^T \boldsymbol{\Phi}_i \boldsymbol{v}|^2 + 3|\tfrac{1}{2}(\boldsymbol{u} + \boldsymbol{v})^T \boldsymbol{\Phi}_i (\boldsymbol{u} + \boldsymbol{v})|^2 \geq |\boldsymbol{u}^T \boldsymbol{\Phi}_i \boldsymbol{v}|^2 \tag{B.28}$$

where we have used the inequality $(a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2$ which follows from convexity of the square. Summing over i, we obtain that

$$\tfrac{3}{4}\|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{u})\|_2^2 + \tfrac{3}{4}\|\mathrm{I\!I}(\boldsymbol{v}, \boldsymbol{v})\|_2^2 + \tfrac{3}{4}\|\mathrm{I\!I}(\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v})\|_2^2 \geq \|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2^2 \tag{B.29}$$

this implies that

$$\tfrac{9}{4} \max\left\{\|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{u})\|_2^2, \|\mathrm{I\!I}(\boldsymbol{v}, \boldsymbol{v})\|_2^2, \|\mathrm{I\!I}(\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v})\|_2^2\right\} \geq \|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2^2 \tag{B.30}$$

Because $\boldsymbol{u}, \boldsymbol{v}$ are unit vectors with $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \leq 0$, we have $\|\boldsymbol{u} + \boldsymbol{v}\|_2 \leq \sqrt{2}$, and so

$$4\kappa^2 \geq \max\left\{\|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{u})\|_2^2, \|\mathrm{I\!I}(\boldsymbol{v}, \boldsymbol{v})\|_2^2, \|\mathrm{I\!I}(\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{u} + \boldsymbol{v})\|_2^2\right\}, \tag{B.31}$$

whence

$$9\kappa^2 \geq \|\mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v})\|_2^2 = (\kappa^{\mathrm{I\!I}})^2, \tag{B.32}$$

which is the claimed inequality. □

Lemma 7. *Assume $\sigma \leq 1/(60\kappa\sqrt{D})$. The objective function $f(\boldsymbol{s}) = -\langle \boldsymbol{s}, \boldsymbol{x} \rangle$ has L–Lipschitz gradient in a $1/\kappa$-neighborhood of $\boldsymbol{s}_\natural$ with probability at least $1 - e^{-D/2}$, where $L = \frac{121}{30}$.*

Proof. On a Riemannian manifold S, the conditions for L-Lipschitz gradient in a subset can be expressed as $\frac{d^2}{dt^2}(f \circ \boldsymbol{\gamma})(t) \leq L$ for all unit-speed geodesics $\boldsymbol{\gamma}(t)$ in the subset [198].

Let $\Delta = 1/\kappa$, and let $\boldsymbol{\gamma}(t)$ be a unit-speed geodesic of S in the neighborhood $B(\boldsymbol{s}_\natural, \Delta)$, $t \in [0, T]$. The neighborhood constraint implies that $T = d(\boldsymbol{\gamma}(0), \boldsymbol{\gamma}(T)) \leq d(\boldsymbol{\gamma}(0), \boldsymbol{s}_\natural) + d(\boldsymbol{\gamma}(T), \boldsymbol{s}_\natural) \leq 2\Delta$.

To bound the second derivative $\frac{d^2}{dt^2}(f \circ \boldsymbol{\gamma})(t)$, we have

$$\begin{aligned}
\frac{d^2}{dt^2}(f \circ \boldsymbol{\gamma})(t) &= -\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{x} \rangle \\
&= -\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(0) \rangle - \left\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{s}_\natural - \boldsymbol{\gamma}(0) \right\rangle - \langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{z} \rangle .
\end{aligned} \tag{B.33}$$

The first term can be bounded as

$$\begin{aligned}
-\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(0) \rangle &= -\left\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) - \int_{t_1=0}^{t} \dot{\boldsymbol{\gamma}}(t_1) dt_1 \right\rangle \\
&= -\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) \rangle + \int_{t_1=0}^{t} \langle \ddot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t_1) \rangle \, dt_1 \\
&= 1 + \int_{t_1=0}^{t} \left\langle \ddot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t) - \int_{t_2=t_1}^{t} \ddot{\boldsymbol{\gamma}}(t_2) dt_2 \right\rangle dt_1 \\
&= 1 - \int_{t_1=0}^{t} \int_{t_2=t_1}^{t} \langle \ddot{\boldsymbol{\gamma}}(t), \ddot{\boldsymbol{\gamma}}(t_2) \rangle \, dt_2 dt_1 \\
&\leq 1 + \kappa^2 \int_{t_1=0}^{t} \int_{t_2=t_1}^{t} dt_2 dt_1 \\
&\leq 1 + \frac{1}{2}\kappa^2 T^2 \\
&\leq 1 + 2\kappa^2 \Delta^2,
\end{aligned} \tag{B.34}$$

where we used $\langle \ddot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) \rangle = -1$ (by differentiating both sides of $\langle \dot{\boldsymbol{\gamma}}(t), \boldsymbol{\gamma}(t) \rangle = 0$) and $\langle \ddot{\boldsymbol{\gamma}}(t), \dot{\boldsymbol{\gamma}}(t) \rangle =$

0.

Hence

$$\frac{d^2}{dt^2}(f \circ \boldsymbol{\gamma})(t) \leq 1 + 2\kappa^2\Delta^2 + \kappa\Delta + \kappa\|z\|. \tag{B.35}$$

Since $\|z\|_2 \leq 2\sigma\sqrt{D}$ with probability at least $1 - e^{-D/2}$, combining this with $\Delta = 1/\kappa$ and $\sigma \leq \frac{1}{60\kappa\sqrt{D}}$, we get with high probability $\frac{d^2}{dt^2}(f \circ \boldsymbol{\gamma})(t) \leq \frac{121}{30}$. □

B.3 Chaining Bounds for the Tangent Bundle Process

In this section, we prove the following lemma, which bounds a crucial Gaussian process that arises in the analysis of gradient descent.

Main Bound for Tangent Bundle Process

Theorem 8. *Suppose that* $\Delta \leq 1/\kappa$*, and set*

$$T^{\max} = \sup\left\{ \langle \boldsymbol{v}, z\rangle \;\middle|\; d_S(\boldsymbol{s}, \boldsymbol{s}_\natural) \leq \Delta,\ \boldsymbol{v} \in T_{\boldsymbol{s}}S,\ \|\boldsymbol{v}\|_2 = 1 \right\}. \tag{B.36}$$

Then with probability at least $1 - 1.6e^{-\frac{x^2}{2\sigma^2}}$*, we have*

$$T^{\max} \leq 12\sigma(\kappa\sqrt{2\pi(d+1)} + \sqrt{\log 12\kappa}) + 30x. \tag{B.37}$$

We prove Theorem 8 below. We directly follow the proof of Theorem 5.29 from [206], establishing a chaining argument while accounting for slight discrepancies and establishing exact constants. The main geometric content of this argument is in Lemma 11, which bounds the size of ε-nets for the tangent bundle.

Proof. Set

$$\mathcal{V} = \left\{ \boldsymbol{v} \;\middle|\; \boldsymbol{v} \in T_{\boldsymbol{s}}S,\ \|\boldsymbol{v}\|_2 = 1,\ d_S(\boldsymbol{s}, \boldsymbol{s}_\natural) \leq \Delta \right\}, \tag{B.38}$$

We first prove that $\mathcal{T} = \{\langle \boldsymbol{v}, z\rangle\}_{v \in \mathcal{V}}$ defines a separable, sub-gaussian process. Take any $v, v' \in$

$\mathcal{V}$.

Then

$$\langle \boldsymbol{v}, z\rangle - \langle \boldsymbol{v'}, z\rangle = \langle \boldsymbol{v} - \boldsymbol{v'}, z\rangle \sim \mathcal{N}(0, \sigma^2 d(\boldsymbol{v}, \boldsymbol{v'})^2), \tag{B.39}$$

immediately satisfying sub-gaussianity. By Lemma 11, there exists an ε-net $\mathcal{N}(\mathcal{V}, d, \epsilon)$ for $\mathcal{V}$ of size at most $N = (12\kappa/\varepsilon)^{2d+1}$. To see separability, let $\mathcal{N}_k = \mathcal{N}(\mathcal{V}, d, 2^{-k})$ be the epsilon net corresponding to $\epsilon = \dfrac{1}{2^k}$. We can construct a countable dense subset of $\mathcal{V}$ by letting

$$\mathcal{N}_\infty = \bigcup_{k=1}^{\infty} \mathcal{N}(\mathcal{V}, d, 2^{-k}). \tag{B.40}$$

Therefore, the existence of a countable dense subset implies separability of $\mathcal{V}$ immediately implying separability of $\mathcal{T}$. Using these facts, we first prove the result in the finite case $|\mathcal{V}| < \infty$, after which we use separability to extend to the infinite case.

Let $|\mathcal{V}| < \infty$ and k_0 be the largest integer such that $2^{-k_0} \geq \operatorname{diam}(\mathcal{V})$. Define $\mathcal{N}_{k_0} = \mathcal{N}(\mathcal{V}, d, 2^{-k_0})$ to be a 2^{-k_0} net of $\mathcal{V}$ with respect to the metric d. Then for all $\boldsymbol{v} \in \mathcal{V}$, there exists $\pi_0(\boldsymbol{v}) \in \mathcal{N}_{k_0}$ such that $d(\boldsymbol{v}, \pi_0(\boldsymbol{v})) < 2^{-k_0}$.

For $k > k_0$, let $\mathcal{N}_k = \mathcal{N}(\mathcal{V}, d, 2^{-k})$ be a 2^{-k} net of $\mathcal{V}$. Subsequently for all $\boldsymbol{v} \in \mathcal{V}$, there exists $\pi_k(\boldsymbol{v}) \in \mathcal{N}_{k_0}$ such that $d(\boldsymbol{v}, \pi_k(\boldsymbol{v})) < 2^{-k}$.

Now fix any $\boldsymbol{v_0} \in \mathcal{V}$. For any $\boldsymbol{v} \in \mathcal{V}$, sufficiently large n yields $\pi_n(\boldsymbol{v}) = \boldsymbol{v}$. Thus,

$$\langle \boldsymbol{v}, z\rangle - \langle \boldsymbol{v_0}, z\rangle = \sum_{k>k_0} \{\langle \pi_k(\boldsymbol{v}), z\rangle - \langle \pi_{k-1}(\boldsymbol{v}), z\rangle\} \tag{B.41}$$

by the telescoping property, implying

$$\sup_{\boldsymbol{v}\in\mathcal{T}} \{\langle \boldsymbol{v}, z\rangle - \langle \boldsymbol{v_0}, z\rangle\} \leq \sum_{k>k_0} \sup_{\boldsymbol{v}\in\mathcal{V}} \{\langle \pi_k(\boldsymbol{v}), z\rangle - \langle \pi_{k-1}(\boldsymbol{v}), z\rangle\}. \tag{B.42}$$

Using the fact that $\mathcal{T}$ is a sub-gaussian process and Lemma 5.2 of [206], we can bound each

individual sum as

$$\mathbb{P}(\sup_{\boldsymbol{v}\in\mathcal{V}}\{\langle\pi_k(\boldsymbol{v}),\boldsymbol{z}\rangle-\langle\pi_{k-1}(\boldsymbol{v}),\boldsymbol{z}\rangle\}\geq 6\times 2^{-k}\sigma\sqrt{\log|\mathcal{N}_k|}+3\times 2^{-k}x_k)\leq e^{-\frac{x_k^2}{2\sigma^2}}. \tag{B.43}$$

By ensuring that all of the sums are simultaneously controlled, we can arrive at the desired bound. We first derive the complement (i.e. there exists one sum which exceeds the desired value)

$$\mathbb{P}(A^c):=\mathbb{P}(\exists k>k_0 \text{ s.t. } \sup_{\boldsymbol{v}\in\mathcal{V}}\{\langle\pi_k(\boldsymbol{v}),\boldsymbol{z}\rangle-\langle\pi_{k-1}(\boldsymbol{v}),\boldsymbol{z}\rangle\}\geq 6\times 2^{-k}\sigma\sqrt{\log|N_k|}+3\times 2^{-k}x_k) \tag{B.44}$$

$$\leq\sum_{k>k_0}\mathbb{P}(\sup_{\boldsymbol{v}\in\mathcal{V}}\{\langle\pi_k(\boldsymbol{v}),\boldsymbol{z}\rangle-\langle\pi_{k-1}(\boldsymbol{v}),\boldsymbol{z}\rangle\}\geq 6\times 2^{-k}\sigma\sqrt{\log|N_k|}+3\times 2^{-k}x_k) \tag{B.45}$$

$$\leq\sum_{k>k_0}e^{-\frac{x_k^2}{2\sigma^2}} \tag{B.46}$$

$$\leq e^{-\frac{x^2}{2\sigma^2}}\sum_{k>0}e^{-k/2}\leq 1.6e^{-\frac{x^2}{2\sigma^2}} \tag{B.47}$$

Now, using corollary 5.25 of [206] and $|\mathcal{N}|\leq(\frac{12\kappa}{\epsilon})^{2d+1}$, we have

$$\sup_{\boldsymbol{v}\in V}\{\langle\boldsymbol{v},\boldsymbol{z}\rangle-\langle\boldsymbol{v_0},\boldsymbol{z}\rangle\}\leq\sum_{k>k_0}\sup_{\boldsymbol{v}\in\mathcal{V}}\{\langle\pi_k(\boldsymbol{v}),\boldsymbol{z}\rangle-\langle\pi_{k-1}(\boldsymbol{v}),\boldsymbol{z}\rangle\} \tag{B.48}$$

$$\leq 6\sum_{k>k_0}2^{-k}\sigma\sqrt{\log|\mathcal{N}_k|}+3\times 2^{-k_0}\sum_{k>0}2^{-k}\sqrt{k}+3\times 2^{-k_0}\sum_{k>0}2^{-k}x \tag{B.49}$$

$$\leq 12\int_0^\infty\sigma\sqrt{\log\mathcal{N}(\mathcal{V},d,\epsilon)}\,d\epsilon+15\text{diam}(\mathcal{V})x \tag{B.50}$$

$$\leq 12\sigma\sqrt{2d+1}\int_0^\infty\sqrt{(\log(\frac{12\kappa}{\epsilon})}\,d\epsilon+15\text{diam}(\mathcal{V})x \tag{B.51}$$

$$=12\sigma\sqrt{2d+1}(\kappa\sqrt{\pi}\text{ erf }(\log 12\kappa)+\sqrt{\log 12\kappa})+15\text{diam}(\mathcal{V})x \tag{B.52}$$

$$\leq 12\sigma(\kappa\sqrt{2\pi(d+1)}+\sqrt{\log 12\kappa})+15\text{diam}(\mathcal{V})x, \tag{B.53}$$

where we have used $2^{-k_0}\leq 2\text{diam}(\mathcal{V})$, $\sum_{k>0}2^{-k}\sqrt{k}\leq 1.35$ and $\sum_{k>0}2^{-k}\leq 1$ in (B.49), and

$\text{erf}\, z = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2/2} dt \leq 1$ in (B.53).

Thus, if A occurs the above equation holds, implying

$$\mathbb{P}[\sup_{\boldsymbol{v} \in \mathcal{V}} \{\langle \boldsymbol{v}, \boldsymbol{z} \rangle - \langle \boldsymbol{v_0}, \boldsymbol{z} \rangle\} \geq 12\sigma(\kappa\sqrt{2\pi(d+1)} + \sqrt{\log 12\kappa}) + 15\text{diam}(\mathcal{V})x] \leq \mathbb{P}(A^c) \leq 1.6e^{-\frac{x^2}{2\sigma^2}} \tag{B.54}$$

Since $\mathcal{T}$ is a separable process, Theorem 5.24 of [206] directly extends the result to infinite/uncountable $\mathcal{T}$. Letting $\langle \boldsymbol{v_0}, \boldsymbol{z} \rangle = 0$ and noting $\text{diam}(\mathcal{V}) = \sup_{\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{V}} ||\boldsymbol{v} - \boldsymbol{v}'||_2 \leq 2$ yields the claim. □

Nets for $B(\boldsymbol{s}_\natural, \Delta)$

Lemma 9. *Suppose that $\Delta < 1/\kappa$. For any $\varepsilon \in (0, ...]$, there exists an ε-net $\widehat{S}$ for $B(\boldsymbol{s}_\natural, \Delta)$ of size $\#\widehat{S} < (12/\varepsilon)^{d+1}$.*

At a high level, the proof of this lemma proceeds as follows: we form an ε_0 net N_0 for $T_{\boldsymbol{s}_\natural} S$, and then set $\widehat{S} = \{\exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}) \mid \boldsymbol{v} \in N_0\}$. We will argue that $\widehat{S}$ is a $C\varepsilon_0$-net for $B(\boldsymbol{s}_\natural, \Delta)$, by arguing that at length scales $\Delta < 1/\kappa$, the distortion induced by the exponential map is bounded. Crucial to this argument is the following lemma on geodesic triangles:

Lemma 10. *Consider $\boldsymbol{v}, \boldsymbol{v}' \in T_{\boldsymbol{s}_\natural} S$, with $\|\boldsymbol{v}\|_2 = \|\boldsymbol{v}'\|_2 < \Delta$. Then if $\angle(\boldsymbol{v}, \boldsymbol{v}') < \frac{1}{\sqrt{3}}$,*

$$d_S\Big(\exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}), \exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}')\Big) \leq \sqrt{6}\, \Delta\, \angle(\boldsymbol{v}, \boldsymbol{v}'). \tag{B.55}$$

This lemma says that the third side of the triangle with vertices $\boldsymbol{s}_\natural, \exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}), \exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}')$ is at most a constant longer than the third side of an analogous triangle in Euclidean space. The proof of this is a direct application of Toponogov's theorem, a fundamental result in Riemannian geometry which allows one to compare triangles in an arbitrary Riemannian manifold whose sectional curvature is lower bounded to triangles in a constant curvature model space, where one can apply concrete trigonometric reasoning.

Proof of Lemma 10. By Lemma 12, the sectional curvatures κ_s of S are uniformly bounded in

terms of the extrinsic geodesic curvature κ:

$$\kappa_s \geq -\kappa^2. \tag{B.56}$$

By Toponogov's theorem [199], the length $d_S\Big(\exp_{s_\natural}(\boldsymbol{v}), \exp_{s_\natural}(\boldsymbol{v}')\Big)$, of the third side of the geodesic triangle $s_\natural, \exp_{s_\natural}(\boldsymbol{v}), \exp_{s_\natural}(\boldsymbol{v}')$ is bounded by the length of the third side of a geodesic triangles with two sides of length $r = \|\boldsymbol{v}\| = \|\boldsymbol{v}'\|$ and angle $\theta = \angle(\boldsymbol{v}, \boldsymbol{v}')$ in the constant curvature model space $M_{-\kappa^2}$. We can rescale, so that this third length is bounded by L/κ, where L is the length of the third side of a geodesic triangle with two sides of length $r\kappa$ and an angle of $\angle(\boldsymbol{v}, \boldsymbol{v}')$, in the hyperbolic space M_{-1}. Using hyperbolic trigonometry (cf Fact 13 and the identity $\cosh^2 t - \sinh^2 t = 1$), we have

$$\cosh L = 1 + \sinh^2(\kappa r) \times \Big(1 - \cos\theta\Big). \tag{B.57}$$

By convexity of sinh over $[0, \infty)$, for $t \in [0, 1]$, we have $\sinh(t) \leq t \sinh(1)$, and $\sinh^2(t) \leq t^2 \sinh^2(1) < \frac{3}{2}t^2$; since $\kappa r < 1$, $\sinh^2(\kappa r) < \frac{3}{2}\kappa^2 r^2$. Since $\cos(t) \geq 1 - t^2$ for all t, we have

$$\cosh L \leq 1 + \tfrac{3}{2}\kappa^2 r^2 \theta^2. \tag{B.58}$$

Using $\kappa r < 1$, for $\theta < \frac{1}{\sqrt{3}}$ we have $\cosh(L) \leq \frac{3}{2} < \cosh(1)$. Noting that for $t \in [0, 1]$,

$$\cosh(t) \geq g(t) = 1 + \tfrac{1}{4}t^2, \tag{B.59}$$

on $s \in [0, \cosh(1)]$, we have $\cosh^{-1}(s) \leq g^{-1}(s) = 2\sqrt{s-1}$, giving

$$L \leq \sqrt{6} \cdot \kappa r \theta. \tag{B.60}$$

Dividing by κ gives the claimed bound. □

Proof of Lemma 9. Form an (angular) ε_0-net N_0 for $\{\boldsymbol{v} \in T_{\boldsymbol{s}_\natural} S \mid \|\boldsymbol{v}\|_2 = 1\}$ satisfying

$$\forall\ \boldsymbol{v} \in T_{\boldsymbol{s}_\natural} S,\ \exists \widehat{\boldsymbol{v}} \in N_0 \text{ with } \angle(\boldsymbol{v}, \widehat{\boldsymbol{v}}) \leq \varepsilon, \tag{B.61}$$

and an ε_0-net

$$N_r = \{0, \varepsilon_0, 2\varepsilon_0, \ldots, \lfloor \Delta/\varepsilon_0 \rfloor\} \tag{B.62}$$

for the interval $[0, \Delta]$. We can take $\#N_0 \leq (3/\varepsilon_0)^d$ and $\#N_r \leq \Delta/\varepsilon_0 \leq 1/\varepsilon_0$. Combine these two to form a net N for $\{\boldsymbol{v} \in T_{\boldsymbol{s}_0} S \mid \|\boldsymbol{v}\|_2 \leq \Delta\}$ by setting

$$N = \bigcup_{r \in N_r} r N_0. \tag{B.63}$$

Note that $\#N \leq (3/\varepsilon_0)^{d+1}$. Let $\widehat{S} = \{\exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}) \mid \boldsymbol{v} \in N\}$. Consider an arbitrary element $\boldsymbol{s}$ of $B(\boldsymbol{s}_\natural, \Delta)$. There exists $\boldsymbol{v} \in T_{\boldsymbol{s}_\natural} S$ such that $\exp_{\boldsymbol{s}_\natural}(\boldsymbol{v}) = \boldsymbol{s}$. Set

$$\bar{\boldsymbol{v}} = \varepsilon_0 \left\lfloor \frac{\|\boldsymbol{v}\|_2}{\varepsilon_0} \right\rfloor \boldsymbol{v}. \tag{B.64}$$

There exists $\widehat{\boldsymbol{v}} \in N$ with $\|\widehat{\boldsymbol{v}}\|_2 = \|\bar{\boldsymbol{v}}\|_2$ and $\angle(\widehat{\boldsymbol{v}}, \bar{\boldsymbol{v}}) \leq \varepsilon_0$. Note that

$$\widehat{\boldsymbol{s}} = \exp_{\boldsymbol{s}_\natural}(\widehat{\boldsymbol{v}}) \in \widehat{S}. \tag{B.65}$$

By Lemma 10, we have

$$\begin{aligned} d_S\left(\boldsymbol{s}, \widehat{\boldsymbol{s}}\right) &\leq d_S\left(\boldsymbol{s}, \exp_{\boldsymbol{s}_\natural}(\bar{\boldsymbol{v}})\right) + d_S\left(\exp_{\boldsymbol{s}_\natural}(\bar{\boldsymbol{v}}), \widehat{\boldsymbol{s}}\right) \\ &\leq \varepsilon_0 + 3\Delta\varepsilon_0 \\ &< 4\varepsilon_0. \end{aligned} \tag{B.66}$$

Setting $\varepsilon_0 = \varepsilon/4$, we obtain that $\widehat{S}$ is an ε-net for $B(\boldsymbol{s}_\natural, \Delta)$. □

Nets for the Tangent Bundle

Lemma 11. *Set*

$$T = \Big\{ \boldsymbol{v} \mid \boldsymbol{v} \in T_{\boldsymbol{s}} S,\ \|\boldsymbol{v}\|_2 = 1,\ d_S(\boldsymbol{s}, \boldsymbol{s}_\natural) \leq \Delta \Big\}, \tag{B.67}$$

Then there exists an ε-net $\widehat{T}$ for T of size

$$\#\widehat{T} \leq \left(\frac{12\kappa}{\varepsilon} \right)^{2d+1}. \tag{B.68}$$

Proof. Let $\widehat{S}$ be the ε_0-net for $B(\boldsymbol{s}_\natural, \Delta)$. By Lemma 9, there exists such a net of size at most $(12/\varepsilon_0)^{d+1}$. For each $\widehat{\boldsymbol{s}} \in \widehat{S}$, form an ε_1-net $N_{\widehat{\boldsymbol{s}}}$ for

$$\Big\{ \boldsymbol{v} \in T_{\widehat{\boldsymbol{s}}} S \mid \|\boldsymbol{v}\|_2 = 1 \Big\}. \tag{B.69}$$

We set

$$\widehat{T} = \bigcup_{\widehat{\boldsymbol{s}} \in \widehat{\boldsymbol{S}}} N_{\widehat{\boldsymbol{s}}}. \tag{B.70}$$

By [207] Lemma 5.2, we can take $\#N_{\widehat{\boldsymbol{s}}} \leq (3/\varepsilon_1)^d$, and so

$$\#\widehat{T} \leq \left(\frac{3}{\varepsilon_1} \right)^d \left(\frac{12}{\varepsilon_0} \right)^{d+1}. \tag{B.71}$$

Consider an arbitrary element $\boldsymbol{v} \in T$. The vector $\boldsymbol{v}$ belongs to the tangent space $T_{\boldsymbol{s}} S$ for some $\boldsymbol{s}$. By construction, there exists $\widehat{\boldsymbol{s}} \in \widehat{S}$ with $d_S(\boldsymbol{s}, \widehat{\boldsymbol{s}}) \leq \varepsilon$. Consider a minimal geodesic γ joining $\boldsymbol{s}$ and $\widehat{\boldsymbol{s}}$. We generate $\bar{\boldsymbol{v}} \in T_{\widehat{\boldsymbol{s}}} S$ by parallel transporting $\boldsymbol{v}$ along γ. Let $\mathcal{P}_{t,0}$ denote this parallel transport. By [208] Lemma 8.5, the vector field $\boldsymbol{v}_t = \mathcal{P}_{t,0} \boldsymbol{v}$ satisfies

$$\frac{d}{dt} \boldsymbol{v}_t = \mathrm{I\!I}\Big(\dot{\gamma}(s), \boldsymbol{v}_t \Big), \tag{B.72}$$

where $\mathrm{I\!I}(\cdot,\cdot)$ is the second fundamental form. So,

$$\mathcal{P}_{t,0}\boldsymbol{v} = \boldsymbol{v} + \int_0^t \mathrm{I\!I}\Big(\dot\gamma(s), \boldsymbol{v}_s\Big)\, ds. \tag{B.73}$$

By Lemma 6, for every s

$$\Big\|\mathrm{I\!I}\Big(\dot\gamma(s), \boldsymbol{v}_s\Big)\Big\| \leq 3\kappa \tag{B.74}$$

and

$$\|\bar{\boldsymbol{v}} - \boldsymbol{v}\| \leq 3\varepsilon_0\kappa. \tag{B.75}$$

By construction, there is an element $\widehat{\boldsymbol{v}}$ of $N_{\widehat{s}}$ with

$$\|\widehat{\boldsymbol{v}} - \bar{\boldsymbol{v}}\| \leq \varepsilon_1, \tag{B.76}$$

and so $\widehat{T}$ is an $\varepsilon_1 + 3\kappa\varepsilon_0$-net for T. Setting $\varepsilon_1 = \varepsilon/4$ and $\varepsilon_0 = \varepsilon/4\kappa$ completes the proof. □

Supporting Results on Geometry

Lemma 12. *For a Riemannian submanifold S of $\mathbb{R}^D$, the sectional curvatures $\kappa_s(\boldsymbol{v}, \boldsymbol{v}')$ are bounded by the extrinsic geodesic curvature κ, as*

$$\kappa_s(\boldsymbol{v}, \boldsymbol{v}') \geq -\kappa^2. \tag{B.77}$$

Proof. Using the Gauss formula (Theorem 8.4 of [208]), the Riemann curvature tensor R_S of S is related to the Riemann curvature tensor $R_{\mathbb{R}^n}$ of the ambient space via

$$\begin{aligned}
\langle R_S(\boldsymbol{u},\boldsymbol{v})\boldsymbol{v},\boldsymbol{u}\rangle &= \langle R_{\mathbb{R}^D}(\boldsymbol{u},\boldsymbol{v})\boldsymbol{v},\boldsymbol{u}\rangle + \langle \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{v}), \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{v})\rangle - \langle \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{u}), \mathrm{I\!I}(\boldsymbol{v},\boldsymbol{v})\rangle \\
&= \langle \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{v}), \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{v})\rangle - \langle \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{u}), \mathrm{I\!I}(\boldsymbol{v},\boldsymbol{v})\rangle \\
&\geq -\langle \mathrm{I\!I}(\boldsymbol{u},\boldsymbol{u}), \mathrm{I\!I}(\boldsymbol{v},\boldsymbol{v})\rangle\,,
\end{aligned} \tag{B.78}$$

where we have used that $R_{\mathbb{R}^D} = 0$ and $\langle \mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v}), \mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{v}) \rangle \geq 0$. Take any $\boldsymbol{v}, \boldsymbol{v}' \in T_{\boldsymbol{s}} S$. The sectional curvature $\kappa_s(\boldsymbol{v}, \boldsymbol{v}')$ satisfies

$$\kappa_s(\boldsymbol{v}, \boldsymbol{v}') = \kappa_s(\boldsymbol{u}, \boldsymbol{u}') = \langle R_S(\boldsymbol{u}, \boldsymbol{u}')\boldsymbol{u}', \boldsymbol{u} \rangle , \tag{B.79}$$

for any orthonormal basis $\boldsymbol{u}, \boldsymbol{u}'$ for $\mathrm{span}(\boldsymbol{v}, \boldsymbol{v}')$. So

$$\kappa_s(\boldsymbol{v}, \boldsymbol{v}') = \langle R_S(\boldsymbol{u}, \boldsymbol{u}')\boldsymbol{u}', \boldsymbol{u} \rangle \geq - \langle \mathrm{I\!I}(\boldsymbol{u}, \boldsymbol{u}), \mathrm{I\!I}(\boldsymbol{u}', \boldsymbol{u}') \rangle \geq -\kappa^2, \tag{B.80}$$

as claimed. □

Fact 13. *For a hyperbolic triangle with side lengths a, b, c and corresponding (opposite) angles A, B, C, we have*

$$\cosh c = \cosh a \cosh b - \sinh a \sinh b \cos C. \tag{B.81}$$

B.4 Proof of Result (4.9)

In this section, we state and prove the other part of our main claims about gradient descent:

Theorem 14. *Suppose that $\boldsymbol{x} = \boldsymbol{s}_\natural + \boldsymbol{z}$, with $T^{\max}(\boldsymbol{z}) < 1/\kappa$. Consider the constant-stepping Riemannian gradient method, with initial point $\boldsymbol{s}^0$ satisfying $d(\boldsymbol{s}^0, \boldsymbol{s}_\natural) < 1/\kappa$, and step size $\tau = \frac{1}{64}$.*

$$d\left(\boldsymbol{s}^{k+1}, \boldsymbol{s}_\natural\right) \;\; \leq \;\; (1 - \varepsilon) \cdot d\left(\boldsymbol{s}^k, \boldsymbol{s}_\natural\right) + CT^{\max}. \tag{B.82}$$

Here, C and ε are positive numerical constants.

Together with Theorem 8, this result shows that gradient descent rapidly converges to a neighborhood of the truth of radius $C\sigma\sqrt{d}$.

Proof. Let

$$\bar{\boldsymbol{s}}_t = \exp\Big(-t \cdot \mathrm{grad}[f](\boldsymbol{s}^k)\Big) \tag{B.83}$$

be a geodesic joining $\boldsymbol{s}^k$ and $\boldsymbol{s}^{k+1}$, with $\bar{\boldsymbol{s}}_0 = \boldsymbol{s}^k$ and $\bar{\boldsymbol{s}}_\tau = \boldsymbol{s}^{k+1}$. Let $f_\natural$ denote a noise-free version of the objective function, i.e.,

$$f_\natural(\boldsymbol{s}) = -\left\langle \boldsymbol{s}, \boldsymbol{s}_\natural \right\rangle, \tag{B.84}$$

and notice that for all s,

$$\mathrm{grad}[f_\natural](\boldsymbol{s}) = \mathrm{grad}[f](\boldsymbol{s}) + P_{T_s S}\boldsymbol{z}. \tag{B.85}$$

Furthermore, following calculations in Lemma 7, on $B(\boldsymbol{s}_\natural, 1/\kappa)$, the Riemannian hessian of $f_\natural$ is bounded as

$$\left\|\mathrm{Hess}[f_\natural](\boldsymbol{s})\right\| \leq 4. \tag{B.86}$$

Using the relationship

$$\mathrm{grad}[f_\natural](\bar{\boldsymbol{s}}_t) = \mathcal{P}_{\bar{\boldsymbol{s}}_t,\bar{\boldsymbol{s}}_0}\mathrm{grad}[f_\natural](\bar{\boldsymbol{s}}_0) + \int_{r=0}^{t} \mathcal{P}_{\bar{\boldsymbol{s}}_t,\bar{\boldsymbol{s}}_r}\mathrm{Hess}[f_\natural](\bar{\boldsymbol{s}}_r)\mathcal{P}_{\bar{\boldsymbol{s}}_r,\bar{\boldsymbol{s}}_0}\mathrm{grad}[f](\bar{\boldsymbol{s}}_0)\, dr, \tag{B.87}$$

where $\mathcal{P}_{\bar{\boldsymbol{s}}_t,\bar{\boldsymbol{s}}_0}$ to denote parallel transport along the curve $\bar{\boldsymbol{s}}_t$, we obtain that

$$\left\|\mathrm{grad}[f_\natural](\bar{\boldsymbol{s}}_t) - \mathcal{P}_{\bar{\boldsymbol{s}}_t,\bar{\boldsymbol{s}}_0}\mathrm{grad}[f_\natural](\bar{\boldsymbol{s}}_0)\right\| \;\leq\; 4t\|\mathrm{grad}[f](\bar{\boldsymbol{s}}_0)\|_2. \tag{B.88}$$

Along the curve $\bar{\boldsymbol{s}}_t$, the distance to $\boldsymbol{s}_\natural$ evolves as

$$\begin{aligned}
\frac{d}{dt} d\Big(\bar{\boldsymbol{s}}_t, \boldsymbol{s}_\natural\Big) \quad &= \quad -\left\langle \mathcal{P}_{t,0}\operatorname{grad}[f](\bar{\boldsymbol{s}}_0), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle \\
&= \left\langle -\operatorname{grad}[f](\bar{\boldsymbol{s}}_t), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle + \left\langle \operatorname{grad}[f](\bar{\boldsymbol{s}}_t) - \mathcal{P}_{t,0}\operatorname{grad}[f](\bar{\boldsymbol{s}}_0), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle \\
&\le \left\langle -\operatorname{grad}[f](\bar{\boldsymbol{s}}_t), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle + \left\langle \operatorname{grad}[f_\natural](\bar{\boldsymbol{s}}_t) - \mathcal{P}_{t,0}\operatorname{grad}[f_\natural](\bar{\boldsymbol{s}}_0), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle + 2T^{\max} \\
&\le \left\langle -\operatorname{grad}[f](\bar{\boldsymbol{s}}_t), \frac{-\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural}{\|\log_{\bar{\boldsymbol{s}}_t} \boldsymbol{s}_\natural\|_2} \right\rangle + 4t\|\operatorname{grad}[f](\bar{\boldsymbol{s}}_0)\| + 2T^{\max} \\
&\le -\tfrac{1}{2} d\Big(\bar{\boldsymbol{s}}_t, \boldsymbol{s}_\natural\Big) + 4t\, d\Big(\bar{\boldsymbol{s}}_0, \boldsymbol{s}_\natural\Big) + (3+4t)T^{\max} \\
&\le -\tfrac{1}{2} d\Big(\bar{\boldsymbol{s}}_t, \boldsymbol{s}_\natural\Big) + \tfrac{1}{16} d\Big(\bar{\boldsymbol{s}}_0, \boldsymbol{s}_\natural\Big) + 4T^{\max}
\end{aligned} \tag{B.89}$$

where we have used Lemma 15. Setting $X_t = d(\bar{\boldsymbol{s}}_t, \boldsymbol{s}_\natural)$, we have

$$\dot{X}_t \le -\tfrac{1}{4} X_t \tag{B.90}$$

whenever $X_t \ge \frac{1}{4} X_0 + 16T^{\max}$. Hence,

$$X_\tau \le \max\Big\{ e^{-\frac{\tau}{4}} X_0, \tfrac{1}{4} X_0 + 16T^{\max} \Big\}, \tag{B.91}$$

and so

$$d\Big(\boldsymbol{s}^{k+1}, \boldsymbol{s}_\natural\Big) \quad \le \quad \exp\big(-\tfrac{1}{256}\big) \cdot d\Big(\boldsymbol{s}^{k}, \boldsymbol{s}_\natural\Big) + 16T^{\max}, \tag{B.92}$$

as claimed.

□

B.4.1 Supporting Lemmas

Lemma 15. *Suppose that $\Delta < 1/\kappa$. For all $\boldsymbol{s} \in B(\boldsymbol{s}_\natural, \Delta)$, we have*

$$\left\langle -\mathrm{grad}[f](\boldsymbol{s}), \frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle \leq -\tfrac{1}{2} d(\boldsymbol{s}, \boldsymbol{s}_\natural) + T^{\max}. \tag{B.93}$$

Proof. Notice that

$$\begin{aligned} \left\langle -\mathrm{grad}[f](\boldsymbol{s}), \frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle &= \left\langle P_{T_{\boldsymbol{s}} S}(\boldsymbol{s}_\natural + \boldsymbol{z}), \frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle \\ &\leq \left\langle P_{T_{\boldsymbol{s}} S} \boldsymbol{s}_\natural, \frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle + T^{\max}. \end{aligned} \tag{B.94}$$

Consider a unit speed geodesic γ joining $\boldsymbol{s}_\natural$ and $\boldsymbol{s}$, with $\gamma(0) = \boldsymbol{s}_\natural$ and $\gamma(t) = \boldsymbol{s}_\natural$. Then

$$\frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} = \dot{\gamma}(t), \tag{B.95}$$

and

$$\begin{aligned} \left\langle P_{T_{\boldsymbol{s}} S} \boldsymbol{s}_\natural, \frac{-\log_{\boldsymbol{s}} \boldsymbol{s}_\natural}{\|\log_{\boldsymbol{s}} \boldsymbol{s}_\natural\|_2} \right\rangle &= \langle \gamma(0), \dot{\gamma}(t) \rangle \\ &= \underbrace{\langle \gamma(t), \dot{\gamma}(t) \rangle}_{\textbf{this term} = 0} - \int_0^t \langle \dot{\gamma}(s), \dot{\gamma}(t) \rangle \, ds \\ &= -t \|\dot{\gamma}(t)\|_2^2 - \int_0^t \int_t^s \langle \ddot{\gamma}(r), \dot{\gamma}(t) \rangle \, dr \, ds \\ &\leq -d(\boldsymbol{s}, \boldsymbol{s}_\natural) + \tfrac{1}{2} \kappa d^2(\boldsymbol{s}, \boldsymbol{s}_\natural). \end{aligned} \tag{B.96}$$

In particular, this term is bounded by $-\frac{1}{2} d(\boldsymbol{s}, \boldsymbol{s}_\natural)$ when $\Delta < 1/\kappa$. □

Lemma 16. *For $\boldsymbol{s} \in B(\boldsymbol{s}_\natural, \Delta)$, we have*

$$\left\| \mathrm{grad}[f](\boldsymbol{s}) \right\| \leq d(\boldsymbol{s}, \boldsymbol{s}_\natural) + T^{\max} \tag{B.97}$$

Proof. Notice that

$$\begin{aligned}
\left\|\text{grad}[f](\boldsymbol{s})\right\| &= \left\|P_{T_{\boldsymbol{s}}S}(\boldsymbol{s}_\natural + \boldsymbol{z})\right\| \\
&\leq \left\|P_{T_{\boldsymbol{s}}S}\boldsymbol{s}_\natural\right\| + T^{\max} \\
&\leq \left\|P_{T_{\boldsymbol{s}}\mathbb{S}^{D-1}}\boldsymbol{s}_\natural\right\| + T^{\max} \\
&= \sin\angle(\boldsymbol{s}, \boldsymbol{s}_\natural) + T^{\max} \\
&\leq d_{\mathbb{S}^{D-1}}(\boldsymbol{s}, \boldsymbol{s}_\natural) + T^{\max}, \\
&\leq d_S(\boldsymbol{s}, \boldsymbol{s}_\natural) + T^{\max}, \qquad \text{(B.98)}
\end{aligned}$$

as claimed. □

B.5 Additional Experimental Details

B.5.1 Gravitational Wave Generation

Below we introduce some details on Gravitational Wave data generation. Synthetic gravitational waveforms are generated with the PyCBC package [209] with masses uniformly drawn from $[20, 50]$ (times solar mass $M_\odot$) and 3-dimensional spins drawn from a uniform distribution over the unit ball, at sampling rate 2048Hz. Each waveform is padded or truncated to 1 second long such that the peak is aligned at the 0.9 second location, and then normalized to have unit ℓ^2 norm. Noise is simulated as iid Gaussian with standard deviation $\sigma = 0.1$. The signal amplitude is constant $a = 1$. The training set contains 100,000 noisy waveforms, the test set contains 10,000 noisy waveforms and pure noise each, and a separate validation set constructed iid as the test set is used to select optimal template banks for MF.

B.5.2 Handwritten Digit Recognition Experiment Setup

The MNIST training set contains 6,131 images of the digit 3. In particular, we create a training set containing 10,000 images of randomly transformed digit 3 from the MNIST training set, and a

test set containing 10,000 images each of randomly transformed digit 3 and other digits from the MNIST test set. We select a random subset of 1,000 embedded points as the quantization $\hat{\Xi}$ of the parameter space, and construct a k-d tree from it to perform efficient nearest neighbor search for kernel interpolation. Parameters of the trainable TpopT are initialized using heuristics based on the Jacobians, step sizes and smoothing levels from the unrolled optimization, similar to the previous experiment. $\boldsymbol{\xi}^0$ is initialized at the center of the embedding space. We use the Adam optimizer with batch size 100 and constant learning rate 10^{-3}.