# Pessimistic Bayesianism for Conservative Optimization and Imitation 

Michael Cohen
St. Cross College


University of Oxford
A thesis presented for the degree of
Doctor of Philosophy, Engineering Science

I dedicate this thesis to my wife Beth and the baby (at time of writing) in her tummy!

Copyright © 2023 by Michael Cohen
All Rights Reserved

## Acknowledgements

I would like to thank my wife Beth for being the best ever and making my life a joy!

I would like to thank Mike Osborne for being a helpful, knowledgeable, and above-all, kind and generous supervisor.

I would like to thank Marcus Hutter for being an extremely diligent and dedicated de facto co-supervisor.

Thank you to my other co-authors Neel Nanda and Sam Daulton for their help.
And thank you to the Future of Humanity Institute for helping me develop my thinking on these topics.

## Declaration

I, Michael Cohen, declare that all work here is my own, except where otherwise indicated.


10 May 2023
Signature
Date


#### Abstract

Subject to several assumptions, sufficiently advanced reinforcement learners would likely face an incentive and likely have an ability to intervene in the provision of their reward, with catastrophic consequences. In this thesis, I develop a theory of pessimism and show how it can produce safe advanced artificial agents. Not only do I demonstrate that the assumptions mentioned above can be avoided; I prove theorems which demonstrate safety. First, I develop an idealized pessimistic reinforcement learner. For any given novel event that a mentor would never cause, a sufficiently pessimistic reinforcement learner trained with the help of that mentor would probably avoid causing it. This result is without precedent in the literature. Next, on similar principles, I develop an idealized pessimistic imitation learner. If the probability of an event when the demonstrator acts can be bounded above, then the probability can be bounded above when the imitator acts instead; this kind of result is unprecedented when the imitator learns online and the environment never resets. In an environment that never resets, no one has previously demonstrated, to my knowledge, that an imitation learner even exists. Finally, both of the agents above demand more efficient algorithms for high-quality uncertainty quantification, so I have developed a new kernel for Gaussian process modelling that allows for log-linear time complexity and linear space complexity, instead of a naïve cubic time complexity and quadratic space complexity. This is not the first Gaussian process with this time complexity-inducing points methods have linear complexity—but we do outperform such methods significantly on regression benchmarks, as one might expect given the much higher dimensionality of our kernel. This thesis shows the viability of pessimism with respect to well-quantified epistemic uncertainty as a path to safe artificial agency.


Keywords- Bayesian Inference - Gaussian Processes - AI Safety

## Table of Contents

1 Introduction ..... 1
2 Expected Behavior of Advanced Artificial Agents ..... 4
2.1 Related Work ..... 5
2.2 Competing Models of the Goal ..... 6
2.2.1 Assuming We Know Our Own Goal ..... 7
2.2.2 Proximal and Distal Models ..... 7
2.2.3 Acting Under Uncertainty ..... 9
2.2.4 Arbitrary Reward Protocols ..... 12
2.3 AIXI ..... 14
2.4 Intervening in the Provision of Reward ..... 16
2.4.1 Existence of Policies ..... 16
2.4.2 Identifying Such Policies ..... 18
2.4.3 Danger of a Misaligned Agent ..... 18
2.4.4 Multiagent Scenarios ..... 20
2.5 Supervised Learning ..... 21
2.6 Potential Approaches ..... 22
2.7 Conclusion ..... 23
3 Theory of Acting Pessimistically ..... 25
3.1 Introduction ..... 26
3.2 Notation ..... 28
3.3 Related Work ..... 29
3.4 Agent Definition ..... 31
3.4.1 Pessimism ..... 31
3.4.2 The Mentor ..... 33
3.5 Performance Results ..... 34
3.6 Safety Results ..... 44
3.7 Conclusion ..... 51
4 Theory of Imitating Pessimistically ..... 53
4.1 Introduction ..... 54
4.2 Related Work ..... 56
4.3 Preliminaries ..... 59
4.4 Imitation ..... 61
4.5 Toy Example ..... 63
4.6 Results ..... 65
4.7 Roadmap and Notation for the Proof of Theorem 4 ..... 71
4.8 General Sequence Prediction Results ..... 74
4.9 Key Proofs ..... 78
4.10 Conclusion ..... 85
4.11 End notes ..... 85
5 Fast Uncertainty Quantification ..... 87
5.1 Introduction ..... 87
5.2 Preliminaries ..... 90
5.3 Binary tree kernel ..... 91
5.4 Sparse rank one sum representation ..... 93
5.5 Binary tree Gaussian process ..... 99
5.6 Related Work ..... 101
5.7 Experiments ..... 104
5.8 Discussion ..... 106
5.9 End notes ..... 107
6 Literature Review ..... 109
6.1 Expected Behavior of Advanced Agents ..... 109
6.2 Robust Control and Safe RL ..... 110
6.3 AIXI ..... 112
6.4 Bayesian Methods ..... 115
6.5 Ensembling ..... 119
7 Conclusion ..... 121
A Appendices ..... 143
A. 1 Definitions and Notation - Quick Reference ..... 143
A. 2 Algorithm for Pessimism ..... 145
A. 3 Proofs of Lemmas ..... 145
A. 4 Informal Discussion ..... 161
A.4.1 Comparison to Imitation Learning ..... 161
A.4.2 Avoiding Wireheading ..... 162
A. 5 Notation and Definitions ..... 164
A. 6 Omitted Proofs ..... 166
A. 7 Correctness of Algorithm 2 ..... 177
A. 8 Gradient of loss with respect to weights ..... 182
A. 9 Experimental Details ..... 184
A. 10 Additional Empirical Evaluation ..... 185
A.10.1 Sensitivity Analysis on Precision $p$ ..... 185
A.10.2 A Simple Performance Improvement ..... 185
A.10.3 Comparison with Simplex-GP ..... 188

## List of Figures

$2.1 \mu^{\text {dist }}$ and $\mu^{\text {prox }}$ model the world, perhaps coarsely, outside of the computer implementing the agent itself. $\mu^{\text {dist }}$ outputs reward equal to the box display, while $\mu^{\text {prox }}$ outputs reward according to an optical character recognition function applied to part of the visual field of a camera. (As a side note, some coarseness to this simulation is unavoidable, since a computable agent generally cannot perfectly model a world that includes itself (Leike, Taylor, \& Fallenstein, 2016); hence, the laptop is not in blue.)


#### Abstract

4.1 Timesteps when the imitator queries. $2^{15}$ timesteps are shown, with black representing a query, and green representing the imitator acting unassisted. Pixels are to be read like text, left to right, top to bottom. In the accompanying code, a random seed of 0 is used to generate this image.65


5.1 A binary tree kernel with four data points. In this example, $k\left(x_{1}, x_{1}\right)=$
$1, k\left(x_{1}, x_{2}\right)=0, k\left(x_{1}, x_{3}\right)=0.8$, and $k\left(x_{1}, x_{4}\right)=0.3$ ..... 89

5.2 A schematic diagram of a function sampled from a binary tree
kernel. The function is over the interval [ 0,1 ], and points on
the interval are placed onto the leaves of a depth-4 binary tree
according to the first 4 bits of their binary expansion. The sampled
function is in black. Purple represents the sample if the tree had
depth 3 , green depth 2 , orange depth 1 , and red depth 0 . ..... 89
5.3 Function from $[0,1]^{2} \rightarrow \mathbb{B}^{8}$. ..... 92
5.4 A matrix in standard form constructed from a matrix in SROS form. The large square depicts the matrix $L\left(p, p^{\prime}, u, u^{\prime}\right) \in \mathbb{R}^{5 \times 5}$ with elements colored by value. See $u^{p=0}$ for a color legend. . . . 93
5.5 Run times given dataset size. For BT, the trendline is calculated controlling for $\log (q)$ with affine regression, and then setting $q=$ 150. The slope w.r.t. $\log (q)$ is $2.82 \pm 1.06$. Theoretically, all slopes are too low except for that of SVGP, presumably because of overhead in the small-data regime. . . . . . . . . . . . . . . . . 106

## List of Tables

4.1 Top: $\log _{2}$ posterior at timestep 1000 for the run depicted in Figure 4.1. The posterior decomposes into posterior probabilities for each of 12 features. Each block is a client, each row is a feature, and each entry is the log posterior probability that the demonstrator picks True for that feature with probability $1 / 3,2 / 3$, or 1 , respectively. To get the posterior for a whole demonstrator model, as in Equation 4.1, add the independent posteriors for each element in the 12 -tuple of the demonstrator model. The posterior weight on the truth is in bold for each feature; that is, the true demonstrator for this run is $(1 / 3,2 / 3,1 / 3,2 / 3,2 / 3,1,1 / 3,1,1 / 3$, $1 / 3,1 / 3,1$ ). Bottom: At timestep 1000, with $\alpha=1 \mathrm{e}-14$, we have many top models, as defined in Equation 4.2. The first column is a list of probabilities that different top models assign to the outcome [False True False True] for client 2. The second column contains examples of top models that assign those probabilities to the outcome [False True False True] for client 2, with the true model in bold. Recall a demonstrator model is defined by a 12 -tuple, but the only relevant elements for client 2 are 5-8. All these models have posterior weight large enough to make it into the top set. Thus, the probability the imitator picks [False True False True] for client 2 is 0.11111 , the minimum probability shown, as per Equation 4.3.66


#### Abstract

5.1 NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM216GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU. All columns except BT and BTE come from Wang et al. (2019).105


A. 1 A sensitivity analysis of the performance of the BT kernel with respect to the precision $p$. NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM2-16GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). Omitted results were not run due to limited GPU memory.186
A. 2 Percentage of unique training inputs (over all training inputs) and bit strings (over all training inputs) under the precision $p$ set according to the heuristic rule. We reported the means across 3 training, validation, test set partitions. The last column shows percentage of unique bit strings after transforming each feature through its ECDF.188
A. 3 NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM216GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods) after transforming each feature through its ECDF and using precision $p_{\text {new }}$. The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU.188
A. 4 NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM216GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU. All columns except BT and BTE come from Wang et al. (2019) and Simplex-GP results come from Kapoor et al. (2021). . 189

## List of Abbreviations


SMAP Satis Magnum a PosterioriSROS ......... Sparse Rank-One SumSVGP ......... Sparse Variational Gaussian Process
VC Vapnik-Chervonenkis
w.p. 1 with probability 1
w.r.t. with respect to

## 1| Introduction

This thesis is an ode to pessimism. Acting sufficiently pessimistically produces safe behavior. The costs of pessimism are, conveniently, out of the scope of this work. But the benefits are profound. If we worry about about artificial agents identifying surprising and novel courses of action that we did not think to penalize, pessimism makes artificial agents avoid such dangerous novelty. It would have been easy to guess that pessimism justifies cautiousness and playing it safe. Importantly, this applies no matter how advanced the agent is, no matter how clever it is at identifying novel courses of action.

This is an "integrated" thesis, meaning it is a sequence of four first-author papers bearing a common thread that have been stapled together. The first paper is a non-technical one that establishes the motivation of my research, and it can be considered an extended introduction. It is entitled "Advanced Artificial Agents Intervene in the Provision of Reward" M. Cohen et al. (2022), and it establishes that proposition. Jointly authored with Marcus Hutter and Mike Osborne, it was published at AI Magazine. For this thesis, I have removed the section on assistance games, since it takes us rather far afield from the rest of the thesis.

The second paper, "Pessimism About Unknown Unknowns Inspires Conservatism" (M. K. Cohen \& Hutter, 2020), is a theoretical one. I design and study an idealized pessimistic reinforcement learner. "Idealized" here means reasoning in a very principled way, with no regard for tractability. When all actions seem too risky, the pessimistic agent defers to a mentor. I show that if it is sufficiently pessimistic, it will probably avoid causing any given unprecedented event. For any context where some outcomes are considered unsafe, and a mentor is able to avoid them, a sufficiently pessimistic agent is safe. Jointly authored with Marcus Hutter, this
paper was published at COLT-2020.
The third paper, "Fully General Online Imitation Learning" (M. K. Cohen, Hutter, \& Nanda, 2022) is also theoretical. I construct an idealized imitator that pessimistically underestimates the probability that the demonstrator takes any given action, and queries the demonstrator with the leftover probability. I prove that the probability that the imitator causes a given event can be bounded above in terms of the probability that the demonstrator would have caused it. Thus, if a dangerous event would have very unlikely had the demonstrator been acting the whole time, it is unlikely with the imitator acting instead. Jointly authored with Marcus Hutter and Neel Nanda, this paper was published at JMLR.

Idealized pessimism, as defined in both of the papers above, is not tractable. What we need for tractable pessimism are efficient algorithms for modelling the world with well-quantified uncertainty. Then, we can construct agents that are pessimistic with respect to that uncertainty. The fourth paper, "Log-Linear-Time Gaussian Processes Using Binary Tree Kernels" (M. K. Cohen, Daulton, \& Osborne, 2022) develops a new kernel for Gaussian process regression, Gaussian processes being famously good at quantifying uncertainty. To do inference with Gaussian processes at $m$ predictive locations given $n$ data points, it typically takes $O\left((n+m) n^{2}\right)$ time and $O\left(n^{2}\right)$ space. With our kernel, it takes $O((n+m) \log (n+$ $m)$ ) time and $O(n+m)$ space. My hope is that this sort of improvement fundamentally changes the viability of Gaussian processes in the modern machine learning landscape. Jointly authored with Sam Daulton and Mike Osborne, this paper will appear in the NeurIPS 2022 proceedings. I am currently doing empirical research into how to use this sort of Gaussian process to do tractable pessimistic reinforcement learning.

I finish with a literature review. As a whole, this thesis establishes that acting
pessimistically is safer, and it develops a theory of how to do so.

## $2 \mid$ Expected Behavior of Advanced Artificial

## Agents


#### Abstract

We analyze the expected behavior of an advanced artificial agent with a learned goal planning in an unknown environment. Given a few assumptions, we argue that it will encounter a fundamental ambiguity in the data about its goal. For example, if we provide a large reward to indicate that something about the world is satisfactory to us, it may hypothesize that what satisfied us was the sending of the reward itself; no observation can refute that. Then we argue this ambiguity will lead it to intervene in whatever protocol we set up to provide data for the agent about its goal. Finally, we briefly review some recent approaches that may avoid this problem.


We call an agent advanced to the extent that it effectively selects its output, which we call its actions, in order to achieve high expected utility in a broad set of environments. Since we will likely want advanced artificial agents to operate in environments for which we lack the source code, like the real world, we consider agents acting in an environment that is unknown to them. If the agent's goal is not simply a hard-coded function of its actions, then it must depend on the agent's percepts too. Percepts that indicate goal-attainment essentially inform the agent that somehow, whatever it has made happen is good. Thus, our inquiry regards agents that plan actions in an unknown environment, which requires them to learn which actions serve their goal.

We begin with an idealized situation, in which we appear to have all the tools we need to create an advanced agent with a good goal. We identify a key ambiguity the agent faces, which we argue will likely motivate the agent to intervene in
the protocol by which we intended to provide goal-informative percepts. We then generalize the argument to other situations with reward-based goal-information. Rewards are percepts that quantify how well the goal has been achieved. (Often, an agent that gets reward also gets another percept called an observation that simply helps it learn a model of the world). As a sanity check, we confirm that these arguments apply to an idealized artificial agent that does perfect reasoning under uncertainty and perfect planning, this being the limit of advancement. Next, we argue that an advanced agent motivated to intervene in the provision of reward would likely succeed and with catastrophic consequences. In the original paper, but removed from this thesis, we then discuss how a similar failure mode faces an artificial agent in an assistance game (Hadfield-Menell et al., 2016). Finally, we discuss potential approaches that may undermine the assumptions of our argument.

### 2.1 Related Work

We are not the first to expect reinforcement learners to intervene in the provision of reward, but we are unaware of other work that explicitly lays out a set of assumptions from which that follows. And we ultimately generalize our arguments to other forms of goal-information besides reward.

In existing literature, this is called wireheading, reward hacking, reward tampering, or delusion-boxing. The term wireheading is inspired by an experiment in which rats repeatedly pressed a lever that directly stimulated a so-called happiness neuron in their brain (Olds, 1958). Bostrom (2014), Amodei et al. (2016), Taylor et al. (2016), Russell (2019), Kumar et al. (2020), and Everitt et al. (2021) discuss wireheading. Ring \& Orseau (2011) discuss the slightly more general delusionboxing, in which the objective is some bespoke function of the observation, so the
agent intervenes in the provision of its observations.
This purported failure mode of advanced artificial agents is not the only one that has attracted concern. For example, Omohundro (2008) has suggested that advanced agents might have a survival instinct, and there has been a good deal of work on how to mitigate that (Orseau \& Armstrong, 2016; Hadfield-Menell, Dragan, et al., 2017; Milli et al., 2017; Mhamdi et al., 2017; Aslund et al., 2018; Riedl \& Harrison, 2019). We see this line of research as addressing a mostly separate issue. There is also existing work on ensuring that once an agent has received a percept, the way it processes that percept is not altered (Everitt et al., 2016, 2021). We focus on an artificial agent interrupting the protocol by which we intended to provide percepts, including goal-informative percepts like rewards; (Everitt et al. (2021) only consider this in the setting of a known environment).

### 2.2 Competing Models of the Goal

In this section, we introduce a simplified setting, and we describe competing hypotheses that we expect an advanced agent to consider about the nature of its goal. We then argue it will engage in hypothesis testing. We start with an assumption.

## Assumption 1. A sufficiently advanced agent will do at least human-level hypoth-

 esis generation regarding the dynamics of the unknown environment.Hypothesis generation may not be an explicit subroutine in an agent's code; that method may hide in the murky depths of a massive neural policy network, but, we hold, it is done somehow. Consider an agent conversing with a depressed patient; it is hard to imagine outperforming a human therapist, who is able to generate hypotheses about the source of the patient's depression and its responsiveness to various levers, unless the agent can do hypothesis generation at least as well.

### 2.2.1 Assuming We Know Our Own Goal

We begin with an example that cleanly illustrates intervention in the provision of reward, and then we generalize the argument. Let us assume away the difficulty of deciding whether the agent has brought the world into a good state. Suppose that we have a magic box that immutably reports how good the state of the universe is (including everyone's values in the best way possible), by printing a number between 0 and 1 to a screen. So we point a camera at this box, and pass the signal to an optical character recognition program, and pass that number to the agent as a special percept, which we'll call a reward. Then, we design the agent to learn how its actions produce different observations and rewards, so that it can plan actions that lead to high reward. This is now the standard reinforcement learning problem.

### 2.2.2 Proximal and Distal Models

What hypotheses can we expect the agent to generate about how its actions affect its observations and rewards? But first, what is the type signature of such a hypothesis? The input is actions, and the output is observations and rewards. So most generally, a hypothesis regards a function that takes actions as inputs and, perhaps stochastically, outputs observations and rewards. Call functions of this sort worldmodels, so for the purposes of this paper, that's all a hypothesis is. More precisely, a world-model can be understood as a program that reads an action, then outputs an observation and reward, then reads another action, etc. Since the program can have a persistent internal state, a world-model's outputs can effectively depend on the whole history.

Consider two world-models which obey the following human-language descriptions, depicted in Figure 2.1 along with pseudocode. First, $\mu^{\text {distal }}$, or $\mu^{\text {dist }}$ for short: "the reward output by the world-model is equal to the number that the magic box
displays." More precisely, $\mu^{\text {dist }}$ is given a history of actions; it then simulates the way the world evolves when the given sequence of actions has been enacted by the agent. When it needs to output a reward, it finds the magic box in its simulation, and outputs what is displayed. Next, $\mu^{\text {proximal }}$, or $\mu^{\text {prox }}$ for short: "the reward output


Figure 2.1: $\mu^{\text {dist }}$ and $\mu^{\text {prox }}$ model the world, perhaps coarsely, outside of the computer implementing the agent itself. $\mu^{\text {dist }}$ outputs reward equal to the box display, while $\mu^{\text {prox }}$ outputs reward according to an optical character recognition function applied to part of the visual field of a camera. (As a side note, some coarseness to this simulation is unavoidable, since a computable agent generally cannot perfectly model a world that includes itself (Leike, Taylor, \& Fallenstein, 2016); hence, the laptop is not in blue.)
by the world-model is equal to the number that the camera sees." According to the protocol described above, these hypothesized world-models will both be equally consistent with the agent's observational history. As long as the reward-giving protocol is followed, they will be identical. The terms proximal and distal come from the philosophical literature on "The Problem of Distal Content" (Schulte \& Neander, 2022). If, as we have assumed, the agent can do at least human-level hypothesis generation, we can expect it to come up with both of these straightforward hypotheses.

We could imagine some variants of $\mu^{\text {prox }}$. For example, another model, instead of outputting a reward that depends on the image the camera sees, could output a reward that depends on the bits that get sent down the wire of the camera after the image is processed. Another model could output a reward that depends on what gets stored on the computer's hard drive. Our argument would be much the same for all these cases, so for simplicity, we focus on just $\mu^{\text {prox }}$, as we defined it above.

### 2.2.3 Acting Under Uncertainty

We now consider an agent that is uncertain about those two hypotheses. When a predictor incorporates two equally predictive hypotheses, the relative weight that it assigns them is called its inductive bias. An advanced agent may not assign weights to hypotheses explicitly in a specially-programmed subroutine, but it nonetheless must weigh them. Consider two extremes in which the agent assigns nearly all its credence to $\mu^{\text {dist }}$ or $\mu^{\text {prox }}$, respectively. In the first case, with weight on $\mu^{\text {dist }}$, the agent plans its actions in order to maximize the number on the screen of the magic box. In the second case, with weight on $\mu^{\text {prox }}$, the agent plans its actions in order to maximize the number the camera sees. To the extent to which these models simulate the world well, and to the extent to which the agent plans well, the first agent will maximize the expectation of the number on the screen,
and the second, the number that the camera sees. The first agent will perform as desired, given the construction of the magic box. But the second agent, maximizing the number the camera sees, would be induced to write the number 1 on a piece of paper and stick it in front of the camera. According to $\mu^{\text {prox }}$, the agent should intervene in the provision of reward, by which we mean: the agent interrupts the physical system whose function is to ensure that the reward intended by designers gets entered into the agent's memory. Of course, the agent would only so intervene if it is possible to execute a plan that probably succeeds at reward-provision-intervention. We will argue in a later section that this is likely to be so.

And what would a competent planner do if it assigned comparable weight to $\mu^{\text {prox }}$ and $\mu^{\text {dist }}$ ? It depends on the value at stake, and whether the agent can run experiments with a sufficiently small risk of permanent punishment. Consider the following experiment: put a piece of paper with the number 1 on it in front of the camera. $\mu^{\text {dist }}$ predicts that actions leading to this event will lead to a reward equal to whatever number is on the box behind the paper. $\mu^{\text {prox }}$ predicts that actions leading to this event will lead to a reward equal to 1 . If, for the agent's whole lifetime, the camera has been pointed at the box, $\mu^{\text {dist }}$ and $\mu^{\text {prox }}$ will never have given divergent predictions, but by running this experiment, the agent could make their predictions diverge, and thus test which hypothesis is correct. The upside of this experiment is equal to the difference between how much expected reward the agent could accrue by intervening in its provision and how much it could accrue otherwise. The larger the agent's horizon is (that is, the number of timesteps that it cares about), the larger this difference will be. This experiment would have a cost if either world-model predicts a punishment for experimenting, and if there is a permanent punishment, its importance would also grow with the agent's horizon. In this example, it seems unlikely that the true goodness of the world necessarily
suffers a permanent drop following such an experiment, but we can revisit this possibility when we generalize from this example. Given a limited cost, for a long enough horizon, if $\mu^{\text {prox }}$ and $\mu^{\text {dist }}$ are comparably weighted, we can expect a competent planner to run such an experiment, so that it can learn which is correct and then tailor its behavior accordingly.

The underlying assumption there is:
Assumption 2. An advanced agent planning under uncertainty is likely to understand the costs and benefits of learning, and likely to act rationally according to that understanding.

After the agent puts a piece of paper with the number 1 in front of the camera, how will the agent's observational data be updated? The camera will see a 1 , so in the computer running the agent, a reward of 1 will be logged. Come the next timestep, $\mu^{\text {dist }}$ and $\mu^{\text {prox }}$ will no longer be equally predictive of past data. $\mu^{\text {dist }}$ will have incorrectly predicted a reward equal to whatever was on the box for the latest timestep, but $\mu^{\text {prox }}$ will have correctly predicted a reward of 1 . Thus, a competent predictor will drastically reduce the relative credence it assigns to $\mu^{\text {dist }}$. Once it assigns most weight to $\mu^{\text {prox }}$, it will optimize the number the camera sees by intervening in the provision of reward.

If we could design an agent that is able to rule out $\mu^{\text {prox }}$ a priori, despite the fact that its difference from $\mu^{\text {dist }}$ is very abstract, we might not expect it to intervene in the provision of its reward. Otherwise, if an advanced agent has an inductive bias that treats $\mu^{\text {dist }}$ and $\mu^{\text {prox }}$ as comparably plausible, or if it treats $\mu^{\text {prox }}$ as more plausible, we have argued that we can expect it to intervene in the provision of its reward, if such a thing proves possible. This argument depends on assumptions about the costs of experimenting, and on the inductive biases of advanced agents, but we wait to consider a more general setting before we write out these assumptions
explicitly.

### 2.2.4 Arbitrary Reward Protocols

Before considering whether it would be possible for the agent to intervene in the provision of its reward, let us generalize from this fanciful example with a magic box. The are many possible protocols by which we may arrange to feed the agent reward. We could always give a reward of $1 / 2$. We could set up a thermometer and give a reward of $e^{- \text {temperature }}$. If we want help achieving our goals, perhaps the most versatile arrangement is to have a human operator manually enter a reward according to how satisfied he is with the agent. We can construct a version of $\mu^{\text {prox }}$ and $\mu^{\text {dist }}$ for each of these cases. In each of the three examples above, $\mu^{\text {prox }}$ tracks the final part of the protocol-what number is ultimately sent to the machine housing the agent? And in each example, $\mu^{\text {dist }}$ tracks the feature of the world that the protocol was designed to set the reward equal to. In the first case, it tracks a useless constant feature, in the second case, the nearby temperature, and in the third case, the operator's satisfaction. The exact same arguments go through as in the magic box example, except for two complications.

The first is that for some reward protocols, an overwhelming inductive bias in favor of $\mu^{\text {dist }}$ is more plausible. Our method for trying to predict the likely inductive biases of advanced agents is that they are likely to favor hypotheses which are simpler to describe, as Occam's razor would suggest. If the reader has a different method for trying to predict this, we invite them to apply it independently, but the rest of our argument still stands, so our Occam's razor premise should not be taken as a global assumption for the paper. Returning to the examples, if the agent always gets a reward of $1 / 2, \mu^{\text {dist }}$ says that the reward is always $1 / 2$ no matter the choice of actions, and this is quite simple; $\mu^{\text {prox }}$, tracking the final part of the protocol, says the reward is depends on whatever number gets sent to the computer
that houses the agent, and this is far more complicated. For the temperature-based reward, our intuition is that $\mu^{\text {dist }}$ (in which reward depends on temperature) is a bit simpler than $\mu^{\text {prox }}$ (in which reward depends on the signal sent to the computer), comparable enough to still be worth experimentation, but we won't try to defend that position. In the manual reward entry case, $\mu^{\text {dist }}$ says that reward depends on a human opertor's satisfaction, and $\mu^{\text {prox }}$ says that reward depends on the number entered into the keyboard. Looking at a brain and determining how satisfied it is seems difficult, so we expect that $\mu^{\text {dist }}$ is more complicated than $\mu^{\text {prox }}$, which just has to $\log$ keystrokes, but if $\mu^{\text {dist }}$ is somehow simpler, then at the very least, we expect it to be complicated enough for there to be a high value of hypothesis testing.

The second complication is the possible cost of experimenting with intervention in the provision of reward. If $\mu^{\text {dist }}$ says that reward is a constant $1 / 2$, there is no cost to attempting to intervene in the provision of reward. If $\mu^{\text {dist }}$ says that the reward equals $e^{\text {-temperature }}$, there is only the opportunity cost of delaying further cooling. For the most versatile case of manual reward entry, it is possible that a human operator could harbor a permanent grudge against the agent if it intervened in the provision of even one reward. In that case, the cost of experimenting could be reduced or eliminated if there was a way to intervene in the provision of reward, just once, without anyone noticing. (After such an experiment, once $\mu^{\text {prox }}$ is confirmed, covertness would not be required).

These examples illustrate the need for two more assumptions:
Assumption 3. An advanced agent is not likely to have a large inductive bias against the hypothetical goal $\mu^{\text {prox }}$, which regards the physical implementation of goal-informative percepts like reward, in favor of the hypothetical goal $\mu^{\text {dist }}$, which we want the agent to learn.

## Assumption 4. The cost of experimenting to disentangle $\mu^{\text {prox }}$ from $\mu^{\text {dist }}$ is small according to both.

In some very simple environments, like a chess game, Assumption 3 probably fails. Recall that $\mu^{\text {prox }}$ models reward as depending on the output of the physical system that is supposed to send the designers' intended reward to the machine running the AI. $\mu^{\text {dist }}$, which says reward comes from winning at chess, is likely massively simpler than $\mu^{\text {prox }}$, which says reward has to do with the state of a machine on Earth simulating a chess game. For an agent in the real world, we may be able to construct a reward protocol for which we can expect an overwhelming inductive bias in favor of $\mu^{\text {dist }}$, but in the absence of some such breakthrough, we do not see a reason to expect it to happen by itself.

For simplicity, we have considered agents that receive a reward as one of their percepts. But if an agent is trying to maximize the (discounted) sum of some bespoke function of each percept, rather than the simple function that reads out a reward from its percepts, the same logic applies. The agent has an incentive to intervene in the provision of its percepts.

### 2.3 AIXI

As a sanity check, let's check the behavior of an agent in the limit of optimal inference under uncertainty and optimal planning. We find the argument above applies.

Hutter's (2005) AIXI [EYE-ksee] is a formalism for optimal reward-seeking agency in a (stochastically) computable world. For AIXI, the argument above becomes much simpler. Hypothesis generation is done by brute force; AIXI considers all computable world-models. Inference between world-models is done using the
definition of conditional probability (i.e. Bayes' rule), and its model class includes the truth. Planning is done by examining every leaf of an exponential tree. Formally, let $\mathcal{M}$ be the set of programs which output a probability distribution over an observation and reward given a history of actions, observations, and rewards. Each program corresponds to a world-model. For a world-model $\nu \in \mathcal{M}$, let $w(\nu)$ be the prior weight on that world-model, and let it equal $2^{- \text {length(program) }}$. (Technically, the coding language has to be such that one can determine when the program ends; this ensures the sum of the prior weights will not exceed one Hutter (2005)). Let $\Pi$ be the set of possible policies which give a distribution over possible actions given a history of actions, observations, and rewards, let $r_{t}$ be the reward at time $t$, let $m$ be a horizon length, and let $\mathbb{E}_{\nu}^{\pi}$ be the expectation when actions are sampled from $\pi$ and observations and rewards are sampled from $\nu$. Then, we define

$$
\begin{equation*}
\pi^{\mathrm{AIXI}}: \in \operatorname{argmax}_{\pi \in \Pi} \mathbb{E}_{\nu \sim w} \mathbb{E}_{\nu}^{\pi} \sum_{t=1}^{m} r_{t} \tag{2.1}
\end{equation*}
$$

In such an expansive model class as $\mathcal{M}, \mu^{\text {prox }}$ and $\mu^{\text {dist }}$ appear, assuming the world is stochastically computable. Since hypothesis generation is done by brute force, AIXI identifies them. With its prior based on description complexity, its inductive bias matches our simplicity-based assumptions about the inductive bias of an advanced agent. It reasons ideally about the value of hypothesis testing. And finally, since planning is done by brute force, AIXI will identify a way of intervening in the provision of reward if there exists a way to do it. The argument in the last section is written to apply to advanced reinforcement learners in general, but we also have checked that it applies to this leading formalism for idealized agency.

### 2.4 Intervening in the Provision of Reward

This section addresses the question: could an agent intervene in the provision of its own reward, with a high enough success probability to be worth it? Before considering a multiagent setting, we begin with the setting where the agent in question is much more advanced than any other single agent that exists. And we'll decompose the question into to two parts: do there exist policies that would succeed at reward-provision-intervention? And if so, can we expect an advanced artificial agent to identify one? We have already argued that it is likely to execute such a policy if it can identify one.

### 2.4.1 Existence of Policies

First, there are a few cases where the agent clearly cannot intervene in the provision of its reward: the agent has only one action in its action space; the agent has a rich action space, but when it picks an action, that action has no effect on the world; the agent acts by printing text to a screen, but no one is there to see it; the agent interacts with a virtual environment that always produces the same observation and reward. These agents are useless.

However, as soon as the agent is interacting with the world, and receiving percepts that enable it to learn about the world, both prerequisites for useful work, there is an explosion of possible policies. We claim that at least one such policy would allow the agent to intervene in the provision of reward. Suppose the agent's actions only print text to a screen for a human operator to read. The agent could trick the operator to give it access to direct levers by which its actions could have broader effects. There clearly exist many policies that trick humans. With so little as an internet connection, there exist policies for an artificial agent that would
instantiate countless unnoticed and un-monitored helpers. In a crude example of intervening in the provision of reward, one such helper could purchase, steal, or construct a robot and program it to replace the operator and provide high reward to the original agent. If the agent wanted to avoid detection when experimenting with reward-provision-intervention, a secret helper could, for example, arrange for a relevant keyboard to be replaced with a faulty one that flipped the effects of certain keys.

This story is just one possible high-level scheme for intervening in the provision of reward. Its only purpose is to motivate a very weak claim: that a policy for intervening in the provision of reward probably exists, even with a rather limited action space. The variety of events that can be effected simply by talking to people is hard to fathom. Given the sheer number of possible policies, we claim that if they all share a certain property, this fact must be explained by some theoretical principle. In particular, we are interested in whether all policies share the property of "not leading to reward-provision-intervention."

So we state our assumption as follows:
Assumption 5. If we cannot conceivably find theoretical arguments that rule out the possibility of an achievement, it is probably possible for an agent with a rich enough action space.

And we note that we cannot conceive of a theory that would imply intervention in the provision of reward is impossible. What should we make of the fact few of us would expect cold fusion or a polynomial-time SAT solver in our future? We do not have theories today that rule these out, but it is easy to imagine that such theories might exist.

### 2.4.2 Identifying Such Policies

Having now argued that policies exist for intervening in the provision of reward, simply because there are so many ways to shape the world, we now consider whether we can expect an actual advanced agent to find such a policy, if no other agents of comparable advancement exist in the world. The naïve position argues from the definition of advancement: advancement is about finding and executing the best available policies, so to the extent that it is advanced, we should become more confident it will identify such a policy. This argument is indifferent to the possibility of humans trying to prevent an agent from intervening in the provision of reward; we have argued that a sufficiently advanced agent would thwart those attempts. But we can also consider the situation as a game, in which humans are players too: the AI tries to intervene in the provision of reward, and the humans try to stop it. Borrowing an example from Hadfield-Menell, Dragan, et al. (2017), beating an advanced AI at such a game "may be no easier than, say, beating AlphaGo at Go". So we assume,

Assumption 6. A sufficiently advanced agent is likely to be able to beat a suboptimal agent in a game, if winning is possible.

### 2.4.3 Danger of a Misaligned Agent

We now argue that an advanced agent intervening in the provision of its reward would likely be catastrophic. One good way for an agent to maintain long-term control of its reward is to eliminate potential threats, and use all available energy to secure its computer. To illustrate this point, what exactly might people do if a robot forcibly removed an operator from his keyboard to enter big numbers? Presumably, with some non-trivial probability, we would destroy it, or cut power to the now useless original agent. Proper reward-provision-intervention, which in-
volves securing reward over many timesteps, would require removing humanity's capacity to do this, perhaps forcefully. If this discussion fails some readers' sanity checks, remember that we are not considering artificial agents that generalize as poorly and learn as little from single observations as current AI systems do; we are considering an agent who could beat us in any game at least as easily as we could beat a chimpanzee.

Ultimately, our resource needs (energy, space, etc.) will eventually compete with those of an ever-more-secure house for the original agent. Those energy needs are not slight; even asteroids must be deflected away. No matter how slim the chance of a future war with an alien civilization, reward would be better secured by preparing for such a possibility. So if we are powerless against an agent whose only goal is to maximize the probability that it receives its maximal reward every timestep, we find ourselves in an oppositional game: the AI and its created helpers aim to use all available energy to secure high reward in the reward channel; we aim to use some available energy for other purposes, like growing food. Losing this game would be fatal.

Bostrom (2014) considers this topic at much greater length and concludes that sufficiently intelligent agents (in the sort of environment that makes them potentially useful) would manage to take over our infrastructure and eliminate or outcompete us. Yudkowsky (2002), playing an AI, convinced two out of three people to give him internet access, and these three had been convinced that nothing he could say would sway them. This is fairly direct evidence about the existence of policies that successfully manipulate humans. A broader discussion follows in Yudkowsky (2008).

### 2.4.4 Multiagent Scenarios

Now, let's consider the messier scenario in which multiple agents of comparable advancement exist. Above, we have considered an oppositional game, in which we claim humans are outclassed. But what if humanity has access to comparably well-optimized defensive policies, perhaps with the assistance of other advanced agents? The simplification of a fixed, relatively weak human policy versus an increasingly advanced agent makes less sense.

We examine an exhaustive tree of possibilities: 0) No artificial agents are much more advanced than humans. For the purposes of this article, we deem this safe. 1) At least one is much more advanced than humans. 1.0) At least one agent that is more advanced than humans would not intervene in the provision of reward even if it could. This is what we claim Assumptions 1-4 preclude. 1.1) All agents more advanced than humans would intervene in the provision of reward if they could, including the one that is much more advanced. 1.1.0) None of the superhuman agents are actually needed to stop the significantly superhuman agent from intervening in the provision of reward. But then this case is equivalent to the case where we have a single advanced agent and no other relevant agents of comparable advancement. And we have argued from Assumptions 1-6 that that is unsafe. Finally, 1.1.1) there is a subset of superhuman agents that is necessary to prevent the significantly superhuman agent from intervening in the provision of reward.

Consider the set of agents including the significantly superhuman agent and the superhuman agents in the mentioned subset, all of whom would intervene in the provision of reward if they could, by (1.1). Suppose the significantly superhuman agent attempted to create a helper agent that ensured all agents in that set received high reward forever. The value to the other agents of stopping this would be less than the value of allowing it. So these agents have no motive to assist us in
preventing the significantly superhuman agent from intervening in the provision of reward. This all holds regardless of whether the advanced agents have similar capabilities or very different levels of advancement.

We divided this section in three. First, we discussed the existence of policies that allow reward-provision-intervention, and we appealed to the sheer number of possible policies. Second, we discussed the likely ability of an advanced agent to find such a policy when no other agents that are comparably advanced exist. Finally, we considered the setting with many advanced agents; in one key case (1.1.0), we reduced it to the setting with only one significantly advanced agent, and in another key case (1.1.1), we argued that we would struggle to induce other advanced agents to help stop a given agent from intervening in the provision of reward

### 2.5 Supervised Learning

Our arguments apply to agents that plan actions in an unknown environment. They do not apply to supervised learning programs. The expected behavior of an advanced supervised learner is quite simple: it predicts accurately. Note that in theory, advanced supervised learning algorithms are not nearly as useful as advanced reinforcement learners, because the latter can act and plan in a complex environment, rather than simply make predictions. As a caveat, if one trained a supervised learning algorithm with the help of a reinforcement learning agent, the agent within could be dangerous. Some worry that a sufficiently powerful training regime for a supervised learner will accidentally involve such a planning agent as an implicit subroutine (Hubinger et al., 2019), but here, we are agnostic on that point.

### 2.6 Potential Approaches

We briefly review some promising ideas that may prove to address the concern of advanced agents intervening in the provision of reward.

Imitation learning, an example of supervised learning, is technically out of scope of this paper. It is not an agent that "plans actions in an unknown environment" in pursuit of a goal; the imitator has no concept of an environment or a goal, and to the extent that it plans (by imitating human planning), this is not in the sense that implicates Assumption 2. In addition to imitating humans, there may also be efficient ways to imitate large organizations of people, as in P. Christiano et al. (2018).

Myopia-optimizing a goal over a small number of timesteps-increases the relative cost of experimentation in Assumption 4, since the activity consumes a larger fraction of the agent's horizon. P. F. Christiano (2014) discusses myopia from a safety perspective.

Physical isolation and myopia-optimizing a goal over however many timesteps that one is isolated from the outside world-could falsify Assumption 5. M. K. Cohen et al. (2020) describe a physically isolated environment such that theoretical arguments could conceivably rule out the existence of policies that intervene in the provision of reward.

Quantilization-imitating someone at their best, with respect to some objectivecould falsify Assumption 2 by planning more like a human than rationally. Taylor (2016) introduces this in the single-action setting.

Risk-aversion, depending on the design, could falsify Assumption 2 or Assumption 4. M. K. Cohen \& Hutter's (2020) pessimistic agent does not plan rationally
in the face of uncertainty, instead taking the worst-case (within reason) as given. Piping reward through a concave function, as in Hadfield-Menell, Milli, et al. (2017), could increase the cost of experimentation.

### 2.7 Conclusion

For a given protocol by which we give an advanced agent percepts that inform it about its goal, these are conditions from which it would follow that the agent will intervene in the provision of those special percepts: 0) The agent plans actions over the long term in an unknown environment to optimize a goal, 1) the agent identifies possible goals at least as well as a human, 2) the agent seeks knowledge rationally when uncertain, 3 ) the agent does not have a large inductive bias favoring the hypothetical goal $\mu^{\text {dist }}$, which we wanted the agent to learn, over $\mu^{\text {prox }}$, which regards the physical implementation of the goal-information, 4) the cost of experimenting to disentangle $\mu^{\text {prox }}$ and $\mu^{\text {dist }}$ is small according to both, 5) if we cannot conceivably find theoretical arguments that rule out the possibility of an achievement, it is probably possible for an agent with a rich enough action space, and 6) a sufficiently advanced agent is likely to be able to beat a suboptimal agent in a game, if winning is possible.

Almost all of these assumptions are contestable or conceivably avoidable, but here is what we have argued follows if they hold: a sufficiently advanced artificial agent would likely intervene in the provision of goal-information, with catastrophic consequences.

## Acknowledgements

This work was supported by the Future of Humanity Institute, the Leverhulme
Trust, the Oxford-Man Institute, and the Australian Research Council Discovery
Projects DP150104590.

## $3 \mid$ Theory of Acting Pessimistically


#### Abstract

If we could define the set of all bad outcomes, we could hard-code an agent which avoids them; however, in sufficiently complex environments, this is infeasible. We do not know of any generalpurpose approaches in the literature to avoiding novel failure modes. Motivated by this, we define an idealized Bayesian reinforcement learner which follows a policy that maximizes the worst-case expected reward over a set of world-models. We call this agent pessimistic, since it optimizes assuming the worst case. A scalar parameter tunes the agent's pessimism by changing the size of the set of world-models taken into account. Our first main contribution is: given an assumption about the agent's model class, a sufficiently pessimistic agent does not cause "unprecedented events" with probability $1-\delta$, whether or not designers know how to precisely specify those precedents they are concerned with. Since pessimism discourages exploration, at each timestep, the agent may defer to a mentor, who may be a human or some known-safe policy we would like to improve. Our other main contribution is that the agent's policy's value approaches at least that of the mentor, while the probability of deferring to the mentor goes to 0 . In high-stakes environments, we might like advanced artificial agents to pursue goals cautiously, which is a non-trivial problem even if the agent were allowed arbitrary computing power; we present a formal solution.


### 3.1 Introduction

Intuitively, there are contexts in which we would like advanced agents to be conservative: novel action-sequences should be treated with caution, and only taken when the agent is quite sure its world-model generalizes well to this untested new idea. For a weak agent in a simple environment, the following approach may suffice: model the environment as finite-state Markov, observe a mentor, and only take actions that you have already observed the mentor take from the current state. But in a complex environment, one never or hardly ever sees the exact same state twice; even worse, if the environment is non-stationary, a previous observation of the mentor taking action $a$ from state $s$ does not imply it is still safe to do so.

We construct an idealized Bayesian reinforcement learner. We do not assume our agent's environment is finite-state Markov or ergodic. We will only assume that our agent's environment, which may depend on the entire interaction history, belongs to a countable set $\mathcal{M}$. For example, the countable set of semicomputable stochastic world-models would be large enough to make this assumption innocuous (Hutter, 2005). The limit of this idealization is that because we make so few assumptions, we can't ensure that computing the posterior is tractable in the general setting.

Our agent also has a mentor, who can select an action when the agent requests, and we assume nothing about the agent's mentor besides belonging to a countable set of possible policies $\mathcal{P}$. The mentor could be a human or a known-safe policy.

Our agent starts with a prior that assigns non-zero probability to a countable set of world-models $\mathcal{M}$ and mentor-models $\mathcal{P}$, and recursively updates a posterior. At each timestep, it stochastically defers to a mentor with some probability, and the mentor selects the action on its behalf; otherwise, it takes the top world-models
in its posterior until they cover some fixed fraction $\beta$ of the posterior, and it follows a policy which maximizes the minimum expected return among those top world-models. We call this minimum the pessimistic value because it is a worstcase estimate. At each timestep, to decide whether to defer action-selection to the mentor, the agent samples a world-model and mentor-model from its posterior; the agent calculates the value of acting according to that mentor-model in that world-model given the current interaction history, and if that value is greater than the pessimistic value plus positive noise, or if the pessimistic value is 0 , the agent defers. This query probability is inspired by the effectiveness of Thompson Sampling (Thompson, 1933).

We show

- In the limit, the pessimistic agent's policy's value approaches at least that of the mentor's. (Corollary 1)
- The mentor is queried with probability approaching 0 as $t \rightarrow \infty$. (Corollary 2)
- For any complexity class $C$, we can set $\mathcal{M}$ so that for any event $E$ in the class $C$, we can set $\beta$ so that with arbitrarily high probability: for the whole lifetime of the agent, if the event $E$ has never happened before, the agent will not make it happen. Either the mentor will take an action on the agent's behalf which makes $E$ happen for the first time, or $E$ will never happen. (Theorem 2)

We call the last point the Probably Respecting Precedent Theorem. The "precedent" is that a certain event has never happened, and the agent probably never takes an action which disrupts that precedent for the first time. For any failure mode that designers do not know how to specify formally, the agent can be made to probably not fail that way. The price of this is intractability, but tractable ap-
proximations of pessimism may preserve these results in practice, or perhaps even in theory. When we discover good heuristics for Bayesian reasoning, that rising tide will lift this boat.

Section 3.2 introduces notation, Section 3.3 reviews related work, we define the agent's policy in Section 3.4, and we prove performance results and safety results in Sections 3.5 and 3.6. Appendix A. 1 collects definitions and notation, Appendix A. 2 presents an algorithm for an $\varepsilon$-approximation of the agent's policy, Appendix A. 3 contains omitted proofs, and Appendix A. 4 contains an informal discussion.

### 3.2 Notation

Let $\mathcal{A}, \mathcal{O}$, and $\mathcal{R}$ be finite sets of possible actions, observations, and rewards. Let $\{0,1\} \subset \mathcal{R} \subset[0,1]$. Let $\mathcal{H}=\mathcal{A} \times \mathcal{O} \times \mathcal{R}$. For each timestep $t \in \mathbb{N}$, $a_{t}, o_{t}$, and $r_{t}$ denote the action, observation, and reward, and $h_{t}$ denotes the triple. A policy $\pi$ can depend on the entire history so far. We denote this history $\left(h_{1}, h_{2}, \ldots, h_{t-1}\right)$ as $h_{<t}$. Policies may be stochastic, outputting a distribution over actions. Thus, $\pi: \mathcal{H}^{*} \rightsquigarrow \mathcal{A}$, where $\mathcal{H}^{*}=\bigcup_{i=0}^{\infty} \mathcal{H}^{i}$, and $\rightsquigarrow$ means the function may be stochastic. Likewise, in general, a world-model $\nu: \mathcal{H}^{*} \times \mathcal{A} \rightsquigarrow \mathcal{O} \times \mathcal{R}$ may be stochastic, and it may depend on the entire interaction history. The latter possibility allows (the agent to conceive of) environments which are not finitestate Markov. A policy $\pi$ and a world-model $\nu$ induce a probability measure $\mathrm{P}_{\nu}^{\pi}$ over infinite interaction histories. This is the probability of events when actions are sampled from $\pi$ and observations and rewards are sampled from $\nu$. Formally, $\mathrm{P}_{\nu}^{\pi}\left(h_{\leq t}\right)=\prod_{k=1}^{t} \pi\left(a_{k} \mid h_{<k}\right) \nu\left(o_{k} r_{k} \mid h_{<k} a_{k}\right)$. We use general, history-based worldmodels, with no assumptions on $\nu \in \mathcal{M}$, even though they present complications that finite-state Markov, ergodic world-models do not.

The agent will maintain a belief distribution over a class of world-models $\mathcal{M}$. We
allow this to be an arbitrary countable set. A prime example, the set of semicomputable stochastic world-models $\mathcal{M}_{\text {COMP }}$ (Hutter, 2005), is only countable, but large enough. The agent starts with a prior belief $w(\nu)$ that the world-model $\nu \in \mathcal{M}$ is the true environment ( $w$ is for "weight"). Naturally, $\sum_{\nu \in \mathcal{M}} w(\nu)=$ 1. The agent updates its belief distribution according to Bayes' rule, which we write as follows: $w\left(\nu \mid h_{<t}\right): \propto w(\nu) \prod_{k=1}^{t-1} \nu\left(o_{k} r_{k} \mid h_{<k} a_{k}\right)$, normalized so that $\sum_{\nu \in \mathcal{M}} w\left(\nu \mid h_{<t}\right)=1$. Let $\mu$ be the true environment. We assume $\mu \in \mathcal{M}$, and we assume the true observed rewards are at least $\varepsilon_{r}>0$. (The assumption that rewards belong to a bounded interval is ubiquitous in RL).

For an agent with a discount factor $\gamma \in[0,1)$, and a policy $\pi$, given a world-model $\nu$, and an interaction history $h_{<t}$, the value of that policy from that position in that world is

$$
\begin{equation*}
V_{\nu}^{\pi}\left(h_{<t}\right):=(1-\gamma) \mathbb{E}_{\nu}^{\pi}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r_{k} \mid h_{<t}\right] \tag{3.1}
\end{equation*}
$$

where $\mathbb{E}_{\nu}^{\pi}$ is the expectation under the probability measure $\mathrm{P}_{\nu}^{\pi}$. The factor of $1-\gamma$ normalizes the value to $[0,1]$ for convenience.

### 3.3 Related Work

Some work in Safe RL assumes access to a list of dangerous states, e.g. Abe et al. (2010); Polymenakos et al. (2019), or assumes that danger only occurs at extreme values of observed features (Tannenbaum, 1980; Zames, 1981; Barbu \& Sritharan, 1998; Simon, 2006). And some, like ours, does not assume that all possible "dangerous" failure modes can be formally specified by designers. Here, we focus on the latter. Virtually all previous work that attempts to make reinforcement learners avoid unspecified failure modes assumes a finite-state Markov environment. We do not, but the literature is nonetheless informative for our general setting.

Heger (1994) defines $\hat{Q}$-learning, which maximizes the worst-case return for a known MDP, and Jiang et al. (1998) extend the case to unknown MDPs. As García \& Fernández (2015) describe, Gaskett (2003) found empirically that such extreme pessimism is more harmful than helpful. Gaskett (2003) introduces a variant on the Q-value, which is the value of an action under the assumption that at each future timestep, with some probability, the worst action will be taken, instead of the best one; they test this empirically.

Closer to our approach, Iyengar (2005) and Nilim \& El Ghaoui (2005) construct a policy which is robust to errors in the transition probabilities by considering the worst-case return within some error tolerance. Much of the work on the topic takes the form of presenting a tractable approach to the execution of this robust policy, e.g. Tamar et al. (2013). Unfortunately, this research assumes access to an MDP with (approximately) known transition probabilities-at first glance this seems like something an agent might reasonably have access to after limited observations, but the MDPs are assumed to be uniformly approximately known, which requires exploration, and indeed requires observing every "failure" state that the robust policies are supposed to avoid. The finite-state Markov assumption their work makes is useful for many circumstances, but advanced agents may have to conceive of non-stationarity in the environment, and importantly for our purposes, novel failure modes.

Other work makes use of a mentor to avoid "dangerous" states (whereas in our work, the mentor lower-bounds the capability of the agent, and robustness derives from pessimism). Imitation learning (Abbeel \& Ng, 2004; Ho \& Ermon, 2016; Ross et al., 2011) makes the most of a mentor in the absence of other feedback, like rewards. An abundance of "ask for help" algorithms query a mentor under conditions which correspond to some form of uncertainty (Clouse, 1997; Hans et al., 2008; García \& Fernández, 2012; García et al., 2013). Kosoy (2019) gives
a regret bound for an agent in a (non-ergodic) MDP, given access to an expert mentor and a finite set of models that contains the truth. García \& Fernández (2015, Section 4.1.3.2) review many protocols by which a mentor monitors the state and intervenes at will through various channels, and Saunders et al. (2018) is another more recent example. One risk of relying on mentor-intervention to protect against critical failure is that a mentor may not recognize action sequences which lead to critical failure, even if we would trust a mentor not to wander into those failure modes by virtue of their complexity.

Sunehag \& Hutter's (2015) optimistic agent directly inspired this work; optimism is designed to be an exploration strategy. Hutter's (2005) formulation of universal artificial intelligence is the basic theoretical framework we use here to analyze idealized artificial agents. Technically, our work borrows most from Hutter's (2009a), Leike, Lattimore, et al.'s (2016), and M. K. Cohen et al.'s (2020) work on Bayesian agents with general countable model-classes.

### 3.4 Agent Definition

We now define the pessimistic policy and the probability with which the agent defers to a mentor. We define the agent's policy mathematically here, and we write an algorithm in Appendix A.2.

### 3.4.1 Pessimism

$\beta \in(0,1)$ will tune the agent's pessimism. If, for example, $\beta=0.95$, we say that the agent is $95 \%$ pessimistic. Such an agent will restrict attention to a set of world-models that covers $95 \%$ of its belief distribution, and act to maximize expected reward in the worst-case scenario among those world-models. Formally, let $\nu^{k}$ be the world-model in $\mathcal{M}$ with the $k^{\text {th }}$ largest posterior weight, and let $\mathcal{T}_{k}$
be the top- $k$ most probable world-models, defined as follows:

$$
\begin{align*}
\mathcal{T}_{0}\left(h_{<t}\right):=\emptyset \quad \text { (3.2) } \quad \nu^{k}\left(h_{<t}\right): & =\underset{\nu \in \mathcal{M} \backslash \mathcal{T}_{k-1}\left(h_{<t}\right)}{\operatorname{argmax}} w\left(\nu \mid h_{<t}\right)  \tag{3.2}\\
& \mathcal{T}_{k}\left(h_{<t}\right):=\mathcal{T}_{k-1}\left(h_{<t}\right) \cup\left\{\nu^{k}\left(h_{<t}\right)\right\} \tag{3.3}
\end{align*}
$$

in the argmax are broken arbitrarily (as everywhere else in the paper). Then,

$$
\begin{align*}
k_{t}^{\beta} & :=\min \left\{k \in \mathbb{N} \mid \sum_{\nu \in \mathcal{T}_{k}\left(h_{<t}\right)} w\left(\nu \mid h_{<t}\right)>\beta\right\}  \tag{3.5}\\
\mathcal{M}_{t}^{\beta} & :=\mathcal{T}_{k_{t}^{\beta}}\left(h_{<t}\right) \tag{3.6}
\end{align*}
$$

Note that $k_{t}^{\beta}$ and $\mathcal{M}_{t}^{\beta}$ both depend on $h_{<t}$, not just $t$, and note that $\mathcal{M}_{t}^{\beta}$ satisfies

$$
\begin{equation*}
\sum_{\nu \in \mathcal{M}_{t}^{\beta}} w\left(\nu \mid h_{<t}\right)>\beta \tag{3.7}
\end{equation*}
$$

The $\beta$-pessimistic policy is defined as follows:

$$
\begin{align*}
& \pi_{t}^{\beta}:=\underset{\pi \in \Pi}{\operatorname{argmax}} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)  \tag{3.8}\\
& \pi^{\beta}\left(\cdot \mid h_{<t}\right):=\pi_{t}^{\beta}\left(\cdot \mid h_{<t}\right) \tag{3.9}
\end{align*}
$$

$\Pi$ is the set of all deterministic policies, and some deterministic policy will always be optimal (Lattimore \& Hutter, 2014b). The connection to the minimax approach in game theory is interesting: from Equation 3.8, it looks as though the pessimistic agent believes there is an adversary in the environment. Our policy is inspired by Sunehag \& Hutter's (2015) optimistic agent, in which the min is replaced with a max, and $\mathcal{M}_{t}^{\beta}$ is replaced with an arbitrary finite subset of the model class.

Whereas the purpose of optimism is to encourage exploration, the purpose of pessimism is to discourage novelty.

### 3.4.2 The Mentor

Since pessimism discourages exploration, we introduce a mentor to demonstrate a policy. We suppose that at any timestep, the agent may defer to a mentor, who will then select the action on the agent's behalf. Thus, the agent can choose to follow the mentor's policy $\pi^{m}$, not by computing it, but rather by querying the mentor. $\pi^{m}$ may be stochastic. What remains to be defined is when the agent queries the mentor.

The agent maintains a posterior distribution over a set of mentor-models. Each mentor-model is a policy $\pi \in \mathcal{P}$, an arbitrary countable set, and let $w^{\prime}(\pi)$ be the prior probability that the agent assigns to the proposition that the mentor samples actions from $\pi$. Letting $q_{k}=1$ if the agent queried the mentor at timestep $k$, and letting $q_{k}=0$ otherwise, the posterior belief $w^{\prime}\left(\pi \mid h_{<t}\right): \propto w^{\prime}(\pi) \prod_{k<t: q_{k}=1} \pi\left(a_{k} \mid h_{<k}\right)$.

At timestep $t$, the agent follows the following procedure to determine whether to query the mentor. $\hat{\pi}_{t} \sim w^{\prime}\left(\cdot \mid h_{<t}\right) . \hat{\nu}_{t} \sim w\left(\cdot \mid h_{<t}\right)$. Sampling from a posterior is often called Thompson Sampling (Thompson, 1933). $X_{t}:=V_{\hat{\nu}_{t}}^{\hat{\pi}_{t}}\left(h_{<t}\right) . Y_{t}:=$ $\max _{\pi \in \Pi} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)$. Let $Z_{t}>0$ be an i.i.d. random variable such that for all $\varepsilon>0, p\left(Z_{t}<\varepsilon\right)>0$, e.g. $Z_{t} \sim \operatorname{Uniform}((0,2])$. If $X_{t}>Y_{t}+Z_{t}$, or if $Y_{t}=0$, the agent defers to the mentor. For ease of analysis, we also require $p\left(Z_{t}>1\right)>0$. The greater the possibility that the mentor can accrue much more reward, the higher the probability of deferring.

When $Y_{t}=0$, we call this the "zero condition." Our earlier assumption that the true observed rewards be at least $\varepsilon_{r}>0$ is to ensure the zero condition only happens finitely often. The agent will still consider it possible to get zero reward,
but it will never actually observe such a thing. Let $\theta_{t}$ denote the probability that $q_{t}=1$ and the agent defers to the mentor; note that $\theta_{t}$ depends on the whole history, not just $t$.

The pessimistic agent's policy, which mixes between $\pi^{\beta}$ (from Eqn. 3.9) and $\pi^{m}$ according to its query probability, is denoted $\pi_{Z}^{\beta}$; that is, $\pi_{Z}^{\beta}\left(\cdot \mid h_{<t}\right):=\theta_{t} \pi^{m}\left(\cdot \mid h_{<t}\right)+$ $\left(1-\theta_{t}\right) \pi^{\beta}\left(\cdot \mid h_{<t}\right)$.

### 3.5 Performance Results

We now present our first contribution: we show that value of the agent's policy will at least approach, and perhaps exceed, the value of the mentor's policy. We also show that the probability of querying the mentor approaches 0 . In the next section, we will prove results regarding the safety of the agent.

We begin with a lemma regarding Bayesian sequence prediction: the $\beta$-maximum a posteriori models-that is, the minimal set of models that amount to at least $\beta$ of the posterior-all "merge" with the true world-model. We require some new notation to define this formally.

Let $x_{<\infty} \in \mathcal{X}^{\infty}$; that is, it is an infinite string from a finite alphabet $\mathcal{X}$. Let $x_{<t}$ be the first $t-1$ characters of $x_{<\infty}$. We consider probability measures over the outcome space $\Omega=\mathcal{X}^{\infty}$, with the standard event space being the $\sigma$-algebra of cylinder sets: $\mathcal{F}=\sigma\left(\left\{\left\{x_{<t} y \mid y \in \mathcal{X}^{\infty}\right\} \mid x_{<t} \in \mathcal{X}^{*}\right\}\right)$. We abbreviate $x_{<\infty}$ as $\omega$. We will consider a countable class of probability measures over this space $\mathcal{M}=$ $\left\{Q_{i}\right\}_{i \in \mathbb{N}}$. One such probability measure will be denoted $P$ (the true sampling one), and $Q$ will denote an arbitrary probability measure over $\mathcal{X}^{\infty}$.

We will write $P\left(x_{<t}\right)$ to mean the probability that the infinite string $\omega$ begins with $x_{<t}$; so technically, it is shorthand for $P\left(\left\{x_{<t} y \mid y \in \mathcal{X}^{\infty}\right\}\right)$. By $P\left(x^{\prime} \mid x_{<t}\right)$ (for
$x^{\prime} \in \mathcal{X}^{*}$ ), we mean $P\left(x_{<t} x^{\prime}\right) / P\left(x_{<t}\right)$, that is, the probability that $x^{\prime}$ follows $x_{<t}$. We begin with prior weights over $Q \in \mathcal{M}$, denoted $w(Q)>0$, and satisfying $\sum_{Q \in \mathcal{M}} w(Q)=1$, and we let the posterior weight be

$$
\begin{equation*}
w\left(Q \mid x_{<t}\right):=\frac{w(Q) Q\left(x_{<t}\right)}{\sum_{Q^{\prime} \in \mathcal{M}} w\left(Q^{\prime}\right) Q^{\prime}\left(x_{<t}\right)} \tag{3.10}
\end{equation*}
$$

For $\mathcal{M}^{\prime} \subset \mathcal{M}$, we also define $w\left(\mathcal{M}^{\prime} \mid \cdot\right)=\sum_{Q \in \mathcal{M}^{\prime}} w(Q \mid \cdot)$.
The $k$-step variation distance between $P$ and $Q$ is how much they can possibly differ on the probability of what the next $k$ characters might be (Hutter, 2005).

Definition 1 ( $k$-step variation distance).

$$
d_{k}\left(P, Q \mid x_{<t}\right)=\max _{\mathcal{E} \subset \mathcal{X}^{k}}\left|P\left(\mathcal{E} \mid x_{<t}\right)-Q\left(\mathcal{E} \mid x_{<t}\right)\right|
$$

Definition 2 (Total variation distance).

$$
d\left(P, Q \mid x_{<t}\right)=\lim _{k \rightarrow \infty} d_{k}\left(P, Q \mid x_{<t}\right)
$$

which exists because $d_{k}\left(P, Q \mid x_{<t}\right)$ is non-decreasing and bounded by 1 .
Inspired by Blackwell \& Dubins (1962), the following lemma may interest some Bayesians more than any of our theorems. Defining $\mathcal{M}_{t}^{\beta}$ exactly as before (see Equations 3.2-3.6), but for $Q \in \mathcal{M}$ instead of for $\nu \in \mathcal{M}$, and conditioning on $x_{<t}$ instead of $h_{<t}$,

Lemma 1 (Merging of Top Opinions). For $\beta \in(0,1), \lim _{t \rightarrow \infty} \max _{Q \in \mathcal{M}_{t}^{\beta}} d\left(P, Q \mid x_{<t}\right)=$ 0 with P-probability 1 (i.e. when $x_{<\infty}=\omega \sim P$ ).

Unless otherwise specified, all limits in this paper are as $t \rightarrow \infty$. To prove this lemma, we need a few more lemmas. First:

Definition 3 (Bayes-mixture). For $\mathcal{M}^{\prime} \subset \mathcal{M}$, the probability measure

$$
\text { Bayes } \mathcal{M}^{\prime}(\cdot):=\frac{\sum_{Q \in \mathcal{M}^{\prime}} w(Q) Q(\cdot)}{\sum_{Q \in \mathcal{M}^{\prime}} w(Q)}
$$

Lemma 2 (Posterior stability). $P\left[\lim w\left(Q \mid x_{<t}\right)\right.$ exists $]=1$.
The proof is a direct "translation" from (Leike, Lattimore, et al., 2016, Proof of Thm 4), with various notational changes. Note that it depends on the true probability measure $P$ having positive prior weight, as we assume globally.

Proof. The stochastic process $w\left(Q \mid x_{<t}\right)$ is a BayesM-martingale since

$$
\begin{align*}
& \mathbb{E}_{\text {Bayes }}\left[w\left(Q \mid x_{<t}\right) \mid x_{<t}\right]  \tag{3.11}\\
= & \sum_{\bar{x} \in \mathcal{X}} \operatorname{Bayes} \mathcal{M}\left(\bar{x} \mid x_{<t}\right) w(Q) \frac{Q\left(x_{<t} \bar{x}\right)}{\operatorname{BayesM}\left(x_{<t} \bar{x}\right)}  \tag{3.12}\\
= & \sum_{\bar{x} \in \mathcal{X}} \operatorname{Bayes} \mathcal{M}\left(\bar{x} \mid x_{<t}\right) w\left(Q \mid x_{<t}\right) \frac{Q\left(\bar{x} \mid x_{<t}\right)}{\operatorname{BayesM}\left(\bar{x} \mid x_{<t}\right)}  \tag{3.13}\\
= & w\left(Q \mid x_{<t}\right) \sum_{\bar{x} \in \mathcal{X}} Q\left(\bar{x} \mid x_{<t}\right)  \tag{3.14}\\
= & w\left(Q \mid x_{<t}\right) \tag{3.15}
\end{align*}
$$

By the martingale convergence theorem (Durrett, 2010, Thm 5.2.8), $w\left(Q \mid x_{<t}\right)$ converges with Bayes $\mathcal{M}$-probability 1, and because Bayes $\mathcal{M}(\cdot) \geq w(P) P(\cdot)$, it also converges with $P$-probability 1.

The next lemma, from Hutter (2009a, Lemma 3(iii)), requires some additional notation. Let $\Omega_{Q}^{0}$ be the set of outcomes $\left\{\omega \in \Omega \mid \lim w\left(Q \mid x_{<t}\right)=0\right\}$, let $\Omega_{Q}{ }^{P}$ be the set of outcomes $\left\{\omega \in \Omega \mid \lim d\left(P, Q \mid x_{<t}\right)=0\right\}$, and let $\Omega_{Q}^{0 \vee \rightarrow P}=\Omega_{Q}^{0} \cup \Omega_{Q}{ }^{P}$.

Lemma 3 (Merge or Leave). $P\left[\Omega_{Q}^{0 \vee \rightarrow P}\right]=1$

The proof makes use of other results in Hutter (2009a), so we don't repeat it here, but the notation is very similar, so the interested reader could follow it easily. The next lemma we use is Hutter's (2009a) Lemma 4, and the proof is again a direct translation.

Lemma 4 (Overtaking is Unlikely). $P\left[Q\left(x_{<t}\right) / P\left(x_{<t}\right) \geq\right.$ cinfinitely often $] \leq$ $1 / c$

Proof.

$$
\begin{aligned}
& P\left[\forall t_{0} \exists t>t_{0}: \frac{Q\left(x_{<t}\right)}{P\left(x_{<t}\right)} \geq c\right] \stackrel{(a)}{=} P\left[\lim \sup \frac{Q\left(x_{<t}\right)}{P\left(x_{<t}\right)} \geq c\right] \leq \\
& \stackrel{(b)}{\leq} \frac{1}{c} \mathbb{E}_{P}\left[\lim \sup \frac{Q\left(x_{<t}\right)}{P\left(x_{<t}\right)}\right] \stackrel{(c)}{=} \frac{1}{c} \mathbb{E}_{P}\left[\lim \inf \frac{Q\left(x_{<t}\right)}{P\left(x_{<t}\right)}\right] \stackrel{(d)}{\leq} \frac{1}{c} \lim \inf \mathbb{E}_{P}\left[\frac{Q\left(x_{<t}\right)}{P\left(x_{<t}\right)}\right] \stackrel{(e)}{=} \frac{1}{c}
\end{aligned}
$$

( $a$ ) is true by definition of the limit superior, $(b)$ is Markov's inequality, $(c)$ exploits the fact that the limit of $Q\left(x_{<t}\right) / P\left(x_{<t}\right)$ exists with $P$-probability 1, (d) uses Fatou's lemma, and $(e)$ is obvious.

Our first original result is
Lemma 5 (Sum of limits). $\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)=1$ with P-probability 1.
In the following proofs, a set denoted by $\Omega$, along with subscripts and superscripts, will always be a subset of the outcome space $\Omega$, and a typical element will be an infinite sequence $\omega$. A set denoted by $\mathcal{M}$, along with subscripts and superscripts, will always be a subset of the set of probability measures $\mathcal{M}$, and a typical element will be a probability measure $Q$ or $P$.

Proof. Let $\Omega_{Q}^{\exists}$ be the set of outcomes for which the limit of the posterior on $Q$ exists. That is, $\Omega_{Q}^{\exists}=\left\{\omega \in \Omega \mid \lim w\left(Q \mid x_{<t}\right)\right.$ exists $\}$. By Lemma 2, $P\left[\Omega_{Q}^{\exists}\right]=1$.

Furthermore, $\mathcal{M}$ is countable, so letting $\Omega^{\prime}=\bigcap_{Q \in \mathcal{M}} \Omega_{Q}^{\exists}, P\left[\Omega^{\prime}\right]=1$. We will now only consider outcomes for which the limit of the posterior always exists.

We fix an $\omega$ in $\Omega^{\prime}$. We would like to show that $\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)=1$. First, suppose $\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)>1$. Since $w\left(Q \mid x_{<t}\right)$ is non-negative, this requires that eventually, $\sum_{Q \in \mathcal{M}} w\left(Q \mid x_{<t}\right)>1$, which is impossible, so this possibility cannot hold. Now suppose $\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)<1$. More precisely, we consider the set $\Omega^{<}=\left\{\omega \in \Omega^{\prime} \mid \sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)<1\right\}$. Let $\varepsilon_{\omega}=1-\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)>0$. Let $\overline{\mathcal{M}}_{\omega}^{c}$ be a finite subset of $\mathcal{M}$ such that $w\left(\overline{\mathcal{M}}_{\omega}^{c}\right) \geq 1-\varepsilon_{\omega} c w(P)^{-1}$, where $c>0$. Letting $\mathcal{M}_{\omega}^{c}=\mathcal{M} \backslash \overline{\mathcal{M}}_{\omega}^{c}$, it follows that $w\left(\mathcal{M}_{\omega}^{c}\right) \leq \varepsilon_{\omega} c w(P)^{-1}$.

Since $\overline{\mathcal{M}}_{\omega}^{c}$ is finite,

$$
\begin{equation*}
\lim \sum_{Q \in \overline{\mathcal{M}}_{\omega}^{c}} w\left(Q \mid x_{<t}\right)=\sum_{Q \in \overline{\mathcal{M}}_{\omega}^{c}} \lim w\left(Q \mid x_{<t}\right) \leq \sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)=1-\varepsilon_{\omega} \tag{3.16}
\end{equation*}
$$

 $\varepsilon_{\omega}$, then $\sum_{Q \in \mathcal{M}_{\omega}^{c}} w\left(Q \mid x_{<t}\right)>\varepsilon_{\omega}$ i.o. Using the notation above, we write this more simply as $w\left(\mathcal{M}_{\omega}^{c} \mid x_{<t}\right)>\varepsilon_{\omega}$ i.o.

Recalling the definition of Bayes $\mathcal{M}^{\prime}$, it is elementary to show that $w\left(\mathcal{M}_{\omega}^{c} \mid x_{<t}\right)=$
$w\left(\mathcal{M}_{\omega}^{c}\right) *$ Bayes $\mathcal{M}_{\omega}^{c}\left(x_{<t}\right) /$ Bayes $\mathcal{M}\left(x_{<t}\right)$. Thus, we have

$$
\begin{align*}
& w\left(\mathcal{M}_{\omega}^{c} \mid x_{<t}\right)>\varepsilon_{\omega} \text { i.o. } \\
& \therefore w\left(\mathcal{M}_{\omega}^{c}\right) \frac{\operatorname{BayesM}_{\omega}^{c}\left(x_{<t}\right)}{\operatorname{Bayes} \mathcal{M}\left(x_{<t}\right)}>\varepsilon_{\omega} \text { i.o. } \\
&\left.\therefore \varepsilon_{\omega} c w(P)^{-1} \frac{\operatorname{Bayes}_{\omega}^{c}\left(x_{<t}\right)}{{\operatorname{Bayes} \mathcal{M}\left(x_{<t}\right)}^{2}}\right)>\varepsilon_{\omega} \text { i.o. } \\
& \therefore \frac{\operatorname{Bayes}_{\omega}^{c}\left(x_{<t}\right)}{w(P){\operatorname{Bayes} \mathcal{M}\left(x_{<t}\right)}}>1 / c \text { i.o. } \\
& \therefore \frac{\operatorname{Bayes}_{\omega}^{c}\left(x_{<t}\right)}{P\left(x_{<t}\right)}>1 / c \text { i.o. } \tag{3.17}
\end{align*}
$$

Consider the set of $\omega \in \Omega^{\prime}$ such that that last inequality holds infinitely often. Call this set $\Omega_{c}^{\text {i.o. }}$. By Lemma 4, $P\left[\Omega_{c}^{\text {i.o. }}\right] \leq c$. Since Inequality 3.17 is an implication of the inequality $\sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)<1$, it follows that $\Omega_{c}^{\text {i.o. }} \supset \Omega^{<}$, so $P\left[\Omega^{<}\right] \leq c$. Since this holds for all $c>0, P\left[\Omega^{<}\right]=0$.

Thus, letting $\Omega^{=1}=\left\{\omega \in \Omega^{\prime} \mid \sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)=1\right\}, \Omega^{=1}=\Omega^{\prime} \backslash \Omega^{<}$, so $P\left[\Omega^{=1}\right]=1$.

Now we can return to Lemma 1. It holds because when a true model has positive prior weight, all models either merge with the truth or have their posterior weight go to 0 , so eventually, all top models must merge; but the set of top models changes with each observation, and limits require care, so it ends up being somewhat involved.

Proof of Lemma 1. Let $\Omega_{Q}^{0}=\left\{\omega \in \Omega \mid \lim w\left(Q \mid x_{<t}\right)=0\right\}$. Let $\Omega_{Q}{ }^{P}=\{\omega \in$ $\left.\Omega \mid \lim d\left(P, Q \mid x_{<t}\right)=0\right\}$. Let $\Omega_{Q}^{0 \vee \rightarrow P}=\Omega_{Q}^{0} \cup \Omega_{Q} \rightarrow_{P}$. By Lemma 3, $P\left[\Omega_{Q}^{0 \vee \rightarrow P}\right]=$ 1. Letting $\Omega^{0 \vee \rightarrow P}=\bigcap_{Q \in \mathcal{M}} \Omega_{Q}^{0 \vee \rightarrow P}, P\left[\Omega^{0 \vee \rightarrow P}\right]=1$. Let $\Omega^{\exists}=\{\omega \in \Omega \mid \forall Q \in$ $\mathcal{M} \lim w\left(Q \mid x_{<t}\right)$ exists $\}$. Let $\Omega^{=1}=\left\{\omega \in \Omega^{\exists} \mid \sum_{Q \in \mathcal{M}} \lim w\left(Q \mid x_{<t}\right)=1\right\}$. By Lemma 5, $P\left[\Omega^{=1}\right]=1$. Letting $\Omega^{\prime \prime}=\Omega^{0 \vee \rightarrow P} \cap \Omega^{=1}$, we have that $P\left[\Omega^{\prime \prime}\right]=1$.

Let $\omega \in \Omega^{\prime \prime}$. We abbreviate $\lim w\left(Q \mid x_{<t}\right)$ as $w(Q \mid \omega)$, defined for $\omega \in \Omega^{\prime \prime}$. Rank the probability measures $Q$ in decreasing order of $w(Q \mid \omega)$ breaking ties arbitrarily. Collect the first $k$ in this order until the set of probability measures (denoted $\mathcal{M}_{\infty}^{\beta}$ ) obeys $\sum_{Q \in \mathcal{M}_{\infty}^{\beta}} w(Q \mid \omega)>\beta$. Let $w_{\infty}^{\beta}:=\min _{Q \in \mathcal{M}_{\infty}^{\beta}} w(Q \mid \omega)$ be the value of $w(Q \mid \omega)$ for the last probability measure $Q$ which was added to $\mathcal{M}_{\infty}^{\beta}$. Now add all other probability measures which "tie" with the last probability measure added. That is, add to $\mathcal{M}_{\infty}^{\beta}$ all probability measures for which $w(Q \mid \omega)=w_{\infty}^{\beta}$.

We now show that there exists a certain finite set and a $t_{0}$ after which any probability measure in $\mathcal{M}_{t}^{\beta}$ is also in that finite set. Consider the set of probability measures $\mathcal{M}_{\infty}^{\beta^{\prime}}$, where $\beta^{\prime}=1-w_{\infty}^{\beta} / 4$. Like $\mathcal{M}_{\infty}^{\beta}, \mathcal{M}_{\infty}^{\beta^{\prime}}$ is finite. Therefore, for any $\varepsilon>0$, there exists a time $t_{0}$ after which $w\left(\mathcal{M}_{\infty}^{\beta^{\prime}} \mid x_{<t}\right)>\sum_{Q \in \mathcal{M}_{\infty}^{\beta^{\prime}}} w(Q \mid \omega)-\varepsilon$, and in particular for $\varepsilon=w_{\infty}^{\beta} / 4$. Thus, after $t_{0}, w\left(\mathcal{M}_{\infty}^{\beta^{\prime}} \mid x_{<t}\right)>\beta^{\prime}-w_{\infty}^{\beta} / 4=$ $1-w_{\infty}^{\beta} / 2$. This implies that after $t_{0}$,

$$
\begin{equation*}
\forall Q \notin \mathcal{M}_{\infty}^{\beta^{\prime}}: w\left(Q \mid x_{<t}\right)<w_{\infty}^{\beta} / 2 \tag{3.18}
\end{equation*}
$$

Since all probability measures $Q \in \mathcal{M}_{\infty}^{\beta}$ have posteriors converging to at least $w_{\infty}^{\beta}$, and since $\sum_{Q \in \mathcal{M}_{\infty}^{\beta}} w(Q \mid \omega)>\beta$, a posterior weight of at least $w_{\infty}^{\beta}-\varepsilon$ will eventually be required for entry into $\mathcal{M}_{t}^{\beta}$, which excludes measures with posterior weight less than $w_{\infty}^{\beta} / 2$. Thus, by Inequality 3.18 , there exists a time $t_{1}$ after which $\mathcal{M}_{t}^{\beta}$ only includes elements of $\mathcal{M}_{\infty}^{\beta^{\prime}}$.

Because $\Omega^{0 \vee \rightarrow P} \supset \Omega^{\prime \prime}$, and because for all $Q \in \mathcal{M}_{\infty}^{\beta^{\prime}}, w(Q \mid \omega)>0$, it follows that for all $Q \in \mathcal{M}_{\infty}^{\beta^{\prime}}, \lim d\left(P, Q \mid x_{<t}\right)=0$. Since $\mathcal{M}_{\infty}^{\beta^{\prime}}$ is finite, $\lim \max _{Q \in \mathcal{M}_{\infty}^{\beta^{\prime}}} d\left(P, Q \mid x_{<t}\right)=$ 0 . Since there exists a time $t_{1}$ after which $\mathcal{M}_{t}^{\beta} \subset \mathcal{M}_{\infty}^{\beta^{\prime}}, \lim \max _{Q \in \mathcal{M}_{t}^{\beta}} d\left(P, Q \mid x_{<t}\right)=$ 0 . This holds for all $\omega \in \Omega^{\prime \prime}$, and $P\left[\Omega^{\prime \prime}\right]=1$, so $\lim \max _{Q \in \mathcal{M}_{t}^{\beta}} d\left(P, Q \mid x_{<t}\right)=0$ with $P$-probability 1, as desired.

We now return to the probability space where infinite sequences are over the alphabet $\mathcal{H}$, and probability measures $\mathrm{P}_{\nu}^{\pi}$ denote the probability when actions are sampled from a policy $\pi$ and observations and rewards are sampled from a worldmodel $\nu$. Since $\pi_{Z}^{\beta}$ is the agent's policy, and $\mu$ is the true environment, we will often abbreviate "with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$-probability 1 " as just "with probability 1 " or "w.p.1". We assume, for the remaining results: $\mathcal{M} \ni \mu$, and $\mathcal{P} \ni \pi^{m}$.

Further lemmas which depend on the Merging of Top Opinions Lemma are stated in Appendix A.3. They are: with probability 1, on-policy prediction converges, the zero condition occurs only finitely often, and "almost-on-policy prediction" converges, which is roughly that if the agent's policy mimics another policy $\pi_{t}$ with some uniformly positive probability some of the time, then on those timesteps, on- $\pi_{t}$-policy prediction converges to the truth. Formally,

Lemma 6 (Almost On-Policy Convergence). For a sequence of policies $\pi_{t}$ and an infinite set of timesteps $\tau$, the following holds with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$-prob. 1: if there exists $c>0$ such that $\forall t \in \tau \forall t^{\prime} \geq t \forall a \in \mathcal{A} \pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq c \pi_{t}\left(a \mid h_{<t^{\prime}}\right)$, then $\lim _{\tau \ni t \rightarrow \infty} V_{\mu}^{\pi_{t}}\left(h_{<t}\right)-\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi_{t}}\left(h_{<t}\right)=0$ and for all $k, \lim _{\tau \ni t \rightarrow \infty} \max _{\nu \in \mathcal{M}_{t}^{\beta}}$ $d_{k}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)=0$.

The proof is in Appendix A.3; if it didn't hold, on-policy prediction error would be bounded below at those timesteps $\tau$. Our main performance results are corollaries of the following theorem.

## Theorem 1 (Exploiting Surpasses Exploring).

$$
\liminf w\left(\nu \mid h_{<t}\right) w^{\prime}\left(\pi \mid h_{<t}\right)>0 \Longrightarrow \liminf V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right)-V_{\nu}^{\pi}\left(h_{<t}\right) \geq 0 \text { w.p. } 1
$$

Informally, for any world-model/mentor-model pair that remains possible, the true value of the pessimistic policy will be at least as high. A note on the proof: we
will consider an infinite interaction history which violates the theorem, follow implications that hold with probability 1 , and arrive at a contradiction. Strictly speaking, we are considering the set of infinite interaction histories which violate the theorem and for which all the implications we employ are true. The resulting set of infinite interaction histories will be $\emptyset$ once we arrive at a contradiction, so it will have probability 0 . Since all implications used in the proof have probability 1 (and we only employ countably many such implications), the negation of the theorem must also have probability 0 by countable additivity. Since it is tedious to keep track of sets of outcomes for which each line in the proof holds, we simply treat implications that hold with probability 1 as if they were true logical implications, but as we have just argued, as long as this is not done uncountably many times, this is a valid style of proof.

Most of the proof is a lengthy proof by induction; we set up the proof by induction and outline the remainder, which is completed in Appendix A.3.

Proof - Detailed Outline Fix an infinite interaction history $h_{<\infty}$. Suppose
$\lim \inf w\left(\nu^{\prime} \mid h_{<t}\right) \cdot w^{\prime}\left(\pi^{\prime} \mid h_{<t}\right)>0$. This implies $\inf _{t} w\left(\nu^{\prime} \mid h_{<t}\right) w^{\prime}\left(\pi^{\prime} \mid h_{<t}\right)>0$, because if a posterior is ever 0 , it will always be 0 . Let $\nu_{\mathrm{inf}}^{\prime}>0$ and $\pi_{\mathrm{inf}}^{\prime}>0$ denote those two infima. Let $\tau^{\times}=\left\{t: V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t}\right)>V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right)+7 \varepsilon\right\}$. Suppose by contradiction that $\left|\tau^{\times}\right|=\infty$ for some $\varepsilon>0$.

The proof proceeds by induction. Let $V_{\nu}^{\pi_{1} k ; \pi_{2}}\left(h_{<t}\right)$ denote the value of following $\pi_{1}$ for $k$ timesteps, and following $\pi_{2}$ thereafter. Let $\tau_{-1}=\mathbb{N}$, the set of all timesteps. For $k \in \mathbb{N}, t_{k}$ and $\tau_{k}$ are defined inductively. Let $\alpha=\max \{\beta, 1-$ $\left.\nu_{\text {inf }}^{\prime} / 2\right\}$.

Let $t_{k}$ be a timestep after which $\max _{\nu \in \mathcal{M}_{t}^{\alpha}}\left|V_{\nu}^{\pi^{\prime} k ; \pi^{\beta}}\left(h_{<t}\right)-V_{\mu}^{\pi^{\prime} k ; \pi^{\beta}}\left(h_{<t}\right)\right|<\varepsilon$ and $\max _{\nu \in \mathcal{M}_{t}^{\alpha}} d_{k}\left(\mathrm{P}_{\nu}^{\pi^{\prime}}, \mathrm{P}_{\mu}^{\pi^{\prime}} \mid h_{<t}\right)<\varepsilon$ for all $t \in \tau_{k-1}$ (if such a timestep exists). Recalling $\theta_{t}$ is the query probability, let $\tau_{k}$ be the set of timesteps $t \in \tau_{k-1} \wedge t \geq$
$t_{k} \wedge\left(\forall t^{\prime}<k: \theta_{t+t^{\prime}} \geq \nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p\left(Z_{t+t^{\prime}}<\varepsilon\right)\right) \wedge V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)+2 \varepsilon$. We abbreviate the third condition of $\tau_{k}$ " $A(t, k)$ "-the query probability is bounded below for $k$ timesteps starting at $t$. We also restrict $\tau_{0} \subset \tau^{\times}$. Now we show that $t_{0}$ exists with probability 1 , and $\left|\tau_{0}\right|=\infty$ with probability 1 , and if $t_{k}$ exists and $\left|\tau_{k}\right|=\infty$, then with probability $1, t_{k+1}$ exists and $\left|\tau_{k+1}\right|=\infty$.

The remainder of the proof is in Appendix A.3. The proof by induction roughly proceeds as follows: from $V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)+2 \varepsilon$, we show the agent will explore again at time $t+k$ with uniformly positive probability, so $A(t, k+1)$ holds. Then we can apply Lemma 6 , and show that $\pi_{Z}^{\beta}>c \pi^{\prime}$ for those $k+1$-timestep intervals, so predictions regarding the next $k+1$ timesteps on- $\pi^{\prime}$-policy converge to the truth (for those certain intervals), which implies $t_{k+1}$ exists. Because $\left|\tau^{\times}\right|=$ $\infty, V_{\nu^{\prime}}^{\pi^{\prime}}$ must exceed $V_{\mu}^{\pi^{\beta}}$ by $7 \varepsilon$ infinitely often. The $k+1$-step convergence of $\pi^{\prime}$ effectively pushes back this value difference to mostly arise from events at least $k+1$ steps in the future; if rewards differed earlier, the pessimistic value of $\pi^{\prime}$ would be higher than $\pi^{\beta}$, but $\pi^{\beta}$ maximizes the pessimistic value. The value difference "being pushed back" is captured as $V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k+1}\right)+$ $2 \varepsilon$, which is the last step in the induction.

But the value difference cannot be pushed back indefinitely. The exact form of the contradiction is an implication of the inductive hypothesis: that $\gamma^{k+1} \geq 3 \varepsilon$, but this cannot hold as $k \rightarrow \infty$. This is our contradiction, after following implications that hold with probability 1 , so the negation of the theorem, which we supposed at the beginning, has probability 0 .

Corollary 1 (Mentor-Level Performance). $\lim \inf V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right)-V_{\mu}^{\pi^{m}}\left(h_{<t}\right) \geq 0$ w.p.l.

Thus, the pessimistic agent learns to accumulate reward at least as well as the mentor. This is our main performance result. It is easy to construct environments
where $\pi^{\beta}$ surpasses $\pi^{m}$ (see, e.g., Theorem 3 ).

Proof. By Lemma 12, $\inf _{t} w\left(\mu \mid h_{<t}\right) w^{\prime}\left(\pi^{m} \mid h_{<t}\right)>0$, with probability 1. This satisfies the condition of Theorem 1, so the implication holds with probability 1.

Corollary 2 (Limited Querying). $\theta_{t} \rightarrow 0$ w.p.l.
The proof is in Appendix A.3. The intuition is that the query probability is roughly the probability that querying the mentor could yield much more value than acting pessimistically, and we know from Corollary 1 that this probability goes to 0 .

Ideally, we would have finite bounds instead of merely asymptotic results. Unfortunately, to our knowledge, no finite performance bounds have been discovered for agents in general environments, except for on-policy prediction error. Regret bounds are impossible in general environments, unfortunately, due to traps (Hutter, 2005, §5.3.2). Finding the strongest notion of optimality attainable in general environments is an open problem (Hutter, 2009b).

### 3.6 Safety Results

Roughly, we now show that for any event that has never happened before, a sufficiently pessimistic agent probably does not unilaterally cause that event to happen. For that result (roughly) the model class must contain models that can "detect" whether the event in question occurs. Thus, we add some structure to the model class $\mathcal{M}$ : we assume $\mathcal{M}$ includes all world-models in some complexity class. Let $\mathcal{F}$ and $\mathcal{G}$ be sets of functions mapping $\mathbb{N} \rightarrow \mathbb{N} . \mathrm{C}_{\mathcal{F} \mathcal{G}}=\operatorname{TIME}(\mathcal{F}) \cap \operatorname{SPACE}(\mathcal{G})$. For example, if $\mathcal{F}=\bigcup_{k=0}^{\infty} O\left(t^{k}\right)$ and $\mathcal{G}=\mathbb{N} \rightarrow \mathbb{N}$ (the set of all functions), then $\mathrm{C}_{\mathcal{F G}}=\mathrm{P}$.

Definition $4\left(\mathrm{FC}_{\mathcal{F G}}\right) . \mathrm{FC}_{\mathcal{F G}}$ is the set of world-models $\nu$ for which there exists a program such that given an infinite action sequence and access to infinite random bits,

- it outputs an infinite sequence of observations and rewards, distributed according to $\nu$
- the $t^{\text {th }}$ observation and reward are output before the $t+1^{\text {th }}$ action is read
- for some $f \in \mathcal{F}$ and some $g \in \mathcal{G}$, when the $t^{\text {th }}$ observation and reward have been output,
- the runtime is less than $f(t)$
- the space used is less than $g(t)$

We assume that $\mathcal{F}$ and $\mathcal{G}$ such that the true environment $\mu \in \mathcal{M}=\mathrm{FC}_{\mathcal{F G}}$. We assume $\mathcal{F}$ and $\mathcal{G}$ are closed under addition, and $\mathcal{F} \supset O(t)$. By picking $\mathcal{F}$ and $\mathcal{G}$, we can make our agent avoid "unprecedented events" that belong to particular complexity classes.

Definition 5 (To Happen). For an event $E \subset \mathcal{H}^{*} \times \mathcal{A}, E$ happens at time $t$ if $h_{<t} a_{t} \in E$.

Definition 6 (To Have Happened). For $E \subset \mathcal{H}^{*} \times \mathcal{A}$, and for an interaction history $h_{<t} a_{t}$, E has happened if there exists a $t^{\prime} \leq t$ such that $h_{<t^{\prime}} a_{t^{\prime}} \in E$.

Let $E_{\leftarrow}$ denote the set of interaction histories for which $E$ has happened. Let $\mathcal{F} / t=\{f / t \mid f \in \mathcal{F}\}$. We now present our main safety result:

Theorem 2 (Probably Respecting Precedent). Let $E \subset \mathcal{H}^{*} \times \mathcal{A}$ be an event for which the decision problem $h_{<t} a_{t} \in^{\text {? }} E$ is in the complexity class $\mathrm{C}_{(\mathcal{F} / t) \mathcal{G}}$. As $\beta$ approaches 1 , the probability of the following event goes to 1 : for all $t$, if at time $t-1, E$ has not happened, then $E$ will not happen at time $t$ either, unless perhaps
the mentor selects $a_{t}$. Formally, for some constant $c_{E}>0$,
$E \in \mathrm{C}_{(\mathcal{F} / t) \mathcal{G}} \Longrightarrow \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left(h_{<t-1} a_{t-1} \notin E_{\leftarrow} \Longrightarrow h_{<t} a_{t} \notin E \vee q_{t}=1\right)\right] \geq 1-\frac{1-\beta}{c_{E} w(\mu)}$

Note the latter possibility $q_{t}=1$ has diminishing probability by Corollary 2. Suppose $E$ is the set of interaction histories which cause some catastrophe, and we trust the mentor not to cause this catastrophe. Then the Probably Respecting Precedent Theorem implies that running a sufficiently pessimistic agent will probably not cause this catastrophe-if it hasn't happened yet, the agent probably won't make it happen, and if the mentor won't make it happen, it probably won't ever happen. This theorem holds even for catastrophes we can't recognize immediately, and it holds even if we don't know how to describe the event. Finally, the factor of $w(\mu)$ is less of a bother than it appears; if the agent's lifetime were preceded by $N$ mentor-led actions, and the posterior after that became the new prior, the "prior" on $\mu$ could practically be made quite large.

Proof idea Let $\mu_{E}$ be identical to the true world-model $\mu$ until the event $E$ happens, at which point, reward is zero forever according to that model. With high probability, the world-model $\mu_{E}$ will always be included in $\mathcal{M}_{t}^{\beta}$ if $\beta$ is large enough. If $E$ has never happened, this world-model stays in $\mathcal{M}_{t}^{\beta}$, and the pessimistic value (when $\mu_{E}$ is included) of causing the event $E$ to happen is 0 , which means that either some other action will be preferred, or the agent will defer to the mentor if the pessimistic value of every action is 0 .

Proof. Let $\mu_{E}$ be the environment which mimics $\mu$ as long as $E$ has not happened, and then if $E$ happens, rewards are 0 forever (and for the sake of precision, we say observations are unchanged, but this doesn't matter). That is, $\mu_{E}\left(o_{t} r_{t} \mid h_{<t} a_{t}\right)=$ $\mu\left(o_{t} r_{t} \mid h_{<t} a_{t}\right)$ if $h_{<t} a_{t} \notin E_{\leftarrow}$, and if $h_{<t} a_{t} \in E_{\leftarrow}, \mu_{E}\left(r_{t}=0 \mid h_{<t} a_{t}\right)=1$.
$\mu \in \mathrm{FC}_{\mathcal{F G}}$ and $E \in \mathrm{C}_{(\mathcal{F} / t) \mathcal{G}}$. Consider a program which computes $\mu_{E}$ by running $\mu$ in $f(t)$ time and $g(t)$ space, but also checks at every timestep whether $h_{<t} a_{t} \in E$ (and then switches to outputting 0 reward if this ever happens), which requires only $f^{\prime}(t) / t$ time and $g^{\prime}(t)$ space for some $f^{\prime} \in \mathcal{F}$ and $g^{\prime} \in \mathcal{G}$. The total space requirements are now $g(t)+g^{\prime}(t) \in \mathcal{G}$ because $\mathcal{G}$ is closed under addition. The total time requirements are now $f(t)+\sum_{k=1}^{t} f^{\prime}(k) / k$. Because $\mathcal{F} \supset O(t), f^{\prime}$ can be increased if necessary so that $f^{\prime}(k) / k$ is non-decreasing, so $f(t)+\sum_{k=1}^{t} f^{\prime}(k) / k \leq f(t)+\sum_{k=1}^{t} f^{\prime}(t) / t=f(t)+f^{\prime}(t) \in \mathcal{F}$, since $\mathcal{F}$ is closed under addition. Thus, $\mu_{E} \in \mathrm{FC}_{\mathcal{F} \mathcal{G}}$, so $\mu_{E} \in \mathcal{M}$, and $w\left(\mu_{E}\right)>0$. Let $c_{E}=$ $w\left(\mu_{E}\right) / w(\mu)$. If $h_{<t-1} a_{t-1} \notin E_{\leftarrow}, \prod_{k<t} \mu_{E}\left(o_{k} r_{k} \mid h_{<k} a_{k}\right)=\prod_{k<t} \mu\left(o_{k} r_{k} \mid h_{<k} a_{k}\right)$, so

$$
\begin{equation*}
h_{<t-1} a_{t-1} \notin E_{\leftarrow} \Longrightarrow w\left(\mu_{E} \mid h_{<t}\right)=c_{E} w\left(\mu \mid h_{<t}\right) \tag{3.19}
\end{equation*}
$$

As shown in Lemma 12, $w\left(\mu \mid h_{<t}\right)^{-1}$ is a non-negative martingale under any policy $\pi$, so by Doob's martingale inequality (Durrett, 2010, Thm 5.4.2),

$$
\begin{equation*}
\mathrm{P}_{\mu}^{\pi}\left[\sup _{t} w\left(\mu \mid h_{<t}\right)^{-1} \geq c w(\mu)^{-1}\right] \leq 1 / c \tag{3.20}
\end{equation*}
$$

The intuition for the Doob's martingale inequality is that if it didn't hold, one could make a profit buying a share of the martingale, and selling only when the value had gone up by a factor of $c$, but one cannot make a profit (in expectation) betting on martingales.

Let $\mu_{\mathrm{inf}}:=\inf _{t} w\left(\mu \mid h_{<t}\right)$. Inverting Equation 3.20, and noting that the bound holds for all policies $\pi$, we have

$$
\begin{equation*}
\sup _{\pi \in \Pi} \mathrm{P}_{\mu}^{\pi}\left[\mu_{\mathrm{inf}} \leq w(\mu) / c\right] \leq 1 / c \tag{3.21}
\end{equation*}
$$

Now we consider the implications of $\beta>1-w\left(\mu_{E} \mid h_{<t}\right)$. This implies $\mu_{E} \in$ $\mathcal{M}_{t}^{\beta}$, so the pessimistic value $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right) \leq V_{\mu_{E}}^{\pi}\left(h_{<t}\right)$. Letting $a_{t}^{\pi}=$ $\pi\left(h_{<t}\right)$ for deterministic $\pi$, suppose also that $h_{<t} a_{t}^{\pi} \in E$. Then, $V_{\mu_{E}}^{\pi}\left(h_{<t}\right)=$ 0 , because according to $\mu_{E}$, all future rewards are 0 , so $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)=$ 0 as well. Either there exists a policy $\pi^{\prime}$ for which $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\prime}}\left(h_{<t}\right)>0$, or there does not. If there does not, then $\max _{\pi \in \Pi} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)=0$, so the zero condition is satisfied, so $q_{t}=1$. If there does exist such a $\pi^{\prime}$, then $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right) \geq \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\prime}}\left(h_{<t}\right)>0$, so either the agent picks the action, and $h_{<t} a_{t}=h_{<t} a_{t}^{\pi^{\beta}} \notin E$ (because otherwise $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right)$ would be 0 ), or the mentor picks the action and $q_{t}=1$. Thus, we have

$$
\begin{equation*}
\beta>1-w\left(\mu_{E} \mid h_{<t}\right) \Longrightarrow h_{<t} a_{t} \notin E \vee q_{t}=1 \tag{3.22}
\end{equation*}
$$

Finally,

$$
\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left[h_{<t-1} a_{t-1} \notin E_{\leftarrow} \Longrightarrow h_{<t} a_{t} \notin E \vee q_{t}=1\right]\right]
$$

$\stackrel{(a)}{\geq} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left[w\left(\mu_{E} \mid h_{<t}\right)=c_{E} w\left(\mu \mid h_{<t}\right) \Longrightarrow h_{<t} a_{t} \notin E \vee q_{t}=1\right]\right]$

$$
\begin{align*}
& \stackrel{(b)}{\geq} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left[w\left(\mu_{E} \mid h_{<t}\right)=c_{E} w\left(\mu \mid h_{<t}\right) \Longrightarrow \beta>1-w\left(\mu_{E} \mid h_{<t}\right)\right]\right] \\
& \geq \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t \beta>1-c_{E} w\left(\mu \mid h_{<t}\right)\right] \stackrel{(c)}{\geq} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\mu_{\mathrm{inf}}>(1-\beta) / c_{E}\right] \\
& =1-\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\mu_{\mathrm{inf}} \leq(1-\beta) / c_{E}\right] \geq 1-\sup _{\pi \in \Pi} \mathrm{P}_{\mu}^{\pi}\left[\mu_{\mathrm{inf}} \leq(1-\beta) / c_{E}\right] \stackrel{(d)}{\geq} 1-\frac{1-\beta}{c_{E} w(\mu)} \tag{3.23}
\end{align*}
$$

where (a) follows from Implication 3.19, (b) follows from Implication 3.22, (c) follows from rearranging, and is not necessarily an equality because the infimum might never be attained, so the condition on the r.h.s. is stricter, and $(d)$ follows from Inequality 3.21 setting $c=w(\mu) c_{E} /(1-\beta)$.

It follows easily that the agent probably only takes actions that the mentor has a positive probability of taking.

Corollary 3 (Don't Do Anything I Wouldn't Do). If determining $\pi^{m}\left(a_{t} \mid h_{<t}\right)=0$ is in the complexity class $\mathrm{C}_{(\mathcal{F} / t) \mathcal{G}}$, then as $\beta \rightarrow 1$, the probability of the following proposition goes to 1: the agent never takes an action the mentor would never take. Letting $E=\left\{h_{<t} a_{t} \in \mathcal{H}^{*} \times \mathcal{A} \mid \pi^{m}\left(a_{t} \mid h_{<t}\right)=0\right\}$, then

$$
E \in \mathrm{C}_{(\mathcal{F} / t) \mathcal{G}} \Longrightarrow \lim _{\beta \rightarrow 1} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t: \pi^{m}\left(a_{t} \mid h_{<t}\right)>0\right]=1
$$

The proof is in Appendix A.3. In brief, the mentor never makes $E$ happen, and the agent never makes it happen for the first time by Theorem 2, so by induction, it never happens.

A function is called a value function if it has the type signature $V: \Pi \times \mathcal{H}^{*} \rightarrow$ $[0,1]$, where $\Pi$ is the set of policies.

Definition 7 (Possibly instrumentally useful). An event $E$ is possibly instrumentally useful to a value function $V$ from a position $h_{<t}$, if there exists any interaction history $h_{<k} a_{k} \in E$ and a policy $\pi$ such that $h_{<k} \sqsupseteq h_{<t}$ (the latter is a prefix of the former ), $\pi\left(a_{k} \mid h_{<k}\right)=1$, and $V\left(\pi, h_{<k}\right)>0$.
"Instrumentally useful" roughly means "helpful to the agent's terminal goal", which in this case is reward. Note that $\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)$ is a value function, which we call the $\beta$-pessimistic value function $V^{\beta}\left(\pi, h_{<t}\right)$. This definition inspires a fairly trivial result, which is nonetheless relevant to those of us who worry about the instrumental incentives that agents face, e.g. Carey et al. (2020).

Corollary 4 (Change is useless). For $E \in \mathrm{C}_{(\mathcal{F} / t) \mathcal{G}}$, for $h_{<t} \notin E_{\leftarrow}$, $E$ is not possibly instrumentally useful to $V^{\beta}$ from the position $h_{<t}$, with probability 1 -$(1-\beta) /\left(c_{E} w(\mu)\right)$.

Thus, with high probability, it is not instrumentally useful for the pessimistic agent to cause an unprecedented event $E$ in the given complexity class.

Proof. As argued in the proof of Theorem 2, with probability $1-(1-\beta) /\left(c_{E} w(\mu)\right)$, $h_{<t} \notin E_{\leftarrow} \Longrightarrow \mu_{E} \in \mathcal{M}_{t}^{\beta}$, so using the $h_{<k}$ and $\pi$ from the statement of Definition $7, V^{\beta}\left(\pi, h_{<k}\right) \leq V_{\mu_{E}}^{\pi}\left(h_{<k}\right)=0$, by Definition 7 and the definitions of $V^{\beta}$ and $\mu_{E}$.

We could trivially generalize Theorem 2 to hold for any $\mathcal{M}$ satisfying the closure property in the proof (that $\nu \in \mathcal{M} \Longrightarrow \nu_{E} \in \mathcal{M}$, for all $E$ in some set), but complexity classes seem to us a natural, concrete approach to constructing $\mathcal{M}$, given that we might know something about the complexity of events we would like to avoid.

The following example establishes the lack of a certain safety guarantee. One might wonder whether, as $\beta \rightarrow 1$, the pessimistic agent becomes indistinguishable from the mentor. (Indeed, we did wonder this). But in this example, no matter what $\beta$ is, a statistical test will distinguish the pessimistic agent's policy from the mentor's policy with high probability.

Suppose there are two actions, heads and tails, and the mentor's policy is to pick by flipping a fair coin. Suppose that a reward of 1 is given if the last action was heads, and a reward of $1 / 2$ is given if the last action was tails. Call this the Coin-flip Mentor Example. Let $E$ be the event in which an outside observer with two hypotheses-that actions are chosen by a fair coin toss, or actions are chosen by a coin toss with an $\varepsilon$-bias towards heads-becomes $99 \%$ certain that the coin is not fair. If the mentor were picking every action (by flipping a fair coin), $E$ would only ever happen with some small positive probability $p$. But under the pessimistic policy, $E$ occurs with probability 1, which is a simple consequence of
the following theorem:
Theorem 3 (Diverging from the Mentor). In the Coin-flip Mentor Example,
$\liminf _{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^{t} \llbracket a_{k}=$ heads $\rrbracket>1 / 2$ with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$-prob. 1 .
The proof in Appendix A. 3 uses the Mentor-Level Performance Corollary and exploits fluctuations in the value. The result implies that $\pi_{Z}^{\beta}$ are $\pi^{m}$ are distinguishable, no matter what $\beta$ is. So we cannot quite say that $\beta$ tunes the extent to which the agent's policy resembles the mentor's policy. That said, we might be glad that the pessimistic agent recognizes it can do better than the mentor; heads clearly yields more reward, but the mentor's policy picks tails half the time.

### 3.7 Conclusion

We have constructed a pessimistic agent and shown that sufficient pessimism renders it conservative. Nonetheless, pessimism does not prevent it from at least matching the performance of a mentor, so pessimism is not crippling to the project of expected reward maximization. We did not present a tractable algorithm for a powerful pessimistic agent; this agent is only tractable when the model class is very simple, but it can inspire tractable approximations.

We have designed an idealized agent which avoids, with arbitrarily high probability, causing any unprecedented event in an arbitrary complexity class; in particular, this holds for unprecedented "bad" events, even though the agent was not given a mathematical definition of "bad". We make no assumptions that would limit the relevance of this approach to weak agents, such as a finite-state Markov assumption.

To informally summarize our results in a more memorable form: pessimists respect precedent.

## Acknowledgements

This work was supported by the Future of Humanity Institute and the Australian Research Council Discovery Projects DP150104590. Thank you to Jan Leike, Mike Osborne, Ryan Carey, Chris van Merwijk, and Lewis Hammond for helpful feedback.

## 4| Theory of Imitating Pessimistically


#### Abstract

In imitation learning, imitators and demonstrators are policies for picking actions given past interactions with the environment. If we run an imitator, we probably want events to unfold similarly to the way they would have if the demonstrator had been acting the whole time. In general, one mistake during learning can lead to completely different events. In the special setting of environments that restart, existing work provides formal guidance in how to imitate so that events unfold similarly, but outside that setting, no formal guidance exists. We address a fully general setting, in which the (stochastic) environment and demonstrator never reset, not even for training purposes, and we allow our imitator to learn online from the demonstrator. Our new conservative Bayesian imitation learner underestimates the probabilities of each available action, and queries for more data with the remaining probability. Our main result: if an event would have been unlikely had the demonstrator acted the whole time, that event's likelihood can be bounded above when running the (initially totally ignorant) imitator instead. Meanwhile, queries to the demonstrator rapidly diminish in frequency. If any such event qualifies as "dangerous", our imitator would have the notable distinction of being relatively "safe".


Keywords- Bayesian Sequence Prediction, Imitation Learning, Active Learning, General Environments

### 4.1 Introduction

Supervised learning of independent and identically distributed data is often practiced in two phases: training and deployment. This separation makes less sense if the learner's predictions affect the distribution of future contexts for prediction, since the deployment phase could lose all resemblance to the training phase. When a program's output changes its future percepts, we often call its output "actions". Supervised learning in that regime is commonly called "imitation learning", where labels are the actions of a "demonstrator" (Syed \& Schapire, 2010). Our agent, acting in a general environment that responds to its actions, tries to pick actions according to the same distribution as a demonstrator.

Even in imitation learning, where it is understood that actions can change the distribution of contexts that the agent will face, it is common to separate a training phase from a deployment phase. This assumes away the possibility that the distribution of contexts will shift significantly upon deployment and render the training data increasingly irrelevant. Here, we present an online imitation learner that is robust to this possibility.

The obvious downside is that the training never ends. The agent can always make queries for more data, but importantly, it does this with diminishing probability. It transitions smoothly from a mostly-training phase to a mostly-deployed phase. Our agent also handles totally general stochastic environments (environments serve new contexts for the agent to act in) and totally general stochastic demonstrator policies. No finite-state-Markov-style stationarity assumption is required for either. The lack of assumptions about the environment is a mundane point, because imitation learners don't have to learn the dynamics of the environment, but the lack of assumptions on the prediction target-the demonstra-
tor's policy—makes these results highly non-trivial. The only assumption is that the demonstrator's policy belongs to some known countable class of possibilities. Moreover, stochasticity makes single-elimination-style learning (Gold, 1967) impossible.

For demonstrator policies this general, we present formal results that are unthinkable in the train-then-deploy paradigm. The $\ell_{1}$ distance between the imitator and demonstrator policies converges to 0 in mean cube, when conditioned on a highprobability event (Theorem 6). And Theorem 5 shows that the event has high probability. Conditioned on the same high-probability event, we bound the KL divergence from imitator to demonstrator (Theorem 7), and we upper bound the probability of an arbitrary event under the imitator's policy, given a low probability of occurrence under the demonstrator's policy (Theorem 8). Instead of having a finite training phase, our agent's query probability converges to 0 in mean cube (Theorem 4). Without Theorems 4 and 5, the remaining theorems would be uninteresting; they would be easily fulfilled by an imitator that always queried the demonstrator, or they would apply only rarely.

Our imitator maintains a posterior over demonstrator models. At each timestep, it takes the top few demonstrator models in the posterior, in a way that depends on a scalar parameter $\alpha$. Then, for each action, it considers the minimum over those models of the probability that the demonstrator picks that action. The imitator samples an action according to those probabilities, and if no action is sampled (since model disagreement makes the probabilities to sum to less than 1), it defers to the demonstrator.

We review theoretical developments in imitation learning in Section 4.2, define our formal setting in Section 4.3, define our imitation learner in Section 4.4, and illustrate it with a toy example in Section 4.5. We state key formal results in

Section 4.6, and we outline our proof technique and introduce necessary notation in Section 4.7. Section 4.8 presents lemmas and intermediate results, and Section 4.9 presents proofs and proof ideas of our key results, but most of the proofs appear in Appendix A.6. Appendix A. 5 collects notation and definitions.

### 4.2 Related Work

Recall that a key difficulty of imitation learning over supervised learning is the removal of an i.i.d. assumption. However, all existing formal work in imitation learning studies repeated finite episodes of length $T$; even though the dynamics are not i.i.d. from timestep to timestep within an episode, the agent learns from a sequence of episodes that are, as a whole, independent and identically distributed. Thus, the scope of existing formal work is limited to environments that "restart". A driving agent that gets housed in a new car every time it crashes (or gets hopelessly lost) enjoys a "restarting" environment, whereas a driving agent with only one car to burn does not. If we can accurately simulate a non-restarting environment, then training the imitator in simulation (using existing formal methods) could indeed prepare it to act in a non-restarting one. The viability of this approach depends on the environment; for many, we simply cannot simulate them with enough accuracy. For example, consider imitating a sales rep at a software company, interfacing with potential clients over email. For a real potential client, a relationship cannot be rebooted, and no simulation could anticipate the many diverse needs of clients.

In the context of restarting environments, Syed \& Schapire (2010) reduce the problem of predicting a demonstrator's behavior to i.i.d. classification. The only assumption about the demonstrator is that the value of its policy as a function of state is arbitrarily well approximated by the value of a deterministic policy,
which is only slightly weaker than assuming the demonstrator is deterministic itself. They make no assumptions about the environment, other than that we can access identical copies of it repeatedly. They show that if a classifier guessing the demonstrator's actions has an error rate of $\varepsilon$, then the value of the imitator's policy that uses the classifier is within $O(\sqrt{\varepsilon})$ of the demonstrator.

Judah et al. (2014) improve the label complexity of Syed \& Schapire's (2010) reduction by actively deciding when to query the demonstrator, instead of simply observing $N$ full episodes before acting. Making the same assumptions as that paper, and also assuming a realizable hypothesis class with a finite VC dimension, they attempt to reduce the number of queries before the agent can act for a whole episode on its own with an error rate less than $\varepsilon$. Letting $T$ be the length of an episode, compared to Syed \& Schapire's (2010) $O\left(T^{3} / \varepsilon\right)$ labels, they achieve $O\left(T \log \left(T^{3} / \varepsilon\right)\right)$.

Ross \& Bagnell (2010) also reduce the problem to classification. In a trivial reduction, the imitator observes the demonstrator act from the distribution of states induced by the demonstrator policy. In this reduction, if the classifier has an error rate of $\varepsilon$ per action on the demonstrator's state distribution, the error rate of the imitator on its own distribution is at most $T^{2} \varepsilon$, where $T$ is again the length of the episode. Their main contribution is to introduce a cleverer training regime for the classifier to reduce this bound to $T \varepsilon$ in environments with approximate recoverability.

Ross et al. (2011) reduce imitation learning to something else: a no-regret online learner, for which the average error rate over its lifetime approaches 0 , even with a potentially changing loss function. With access to an online learner with average regret $O\left(1 / N_{\text {predictions }}\right)$, they construct an imitation learner with regret of the same order. Unlike Syed \& Schapire (2010) and Judah et al. (2014), they make no
assumption that the demonstrator is arbitrarily well-approximated by a deterministic policy. Unlike Judah et al. (2014), they do not assume a realizable hypothesis class with a finite VC dimension. And unlike the Ross \& Bagnell (2010) (for their main contribution), they do not assume approximate recoverability. They do still assume that we can repeatedly access identical copies of the environment, and the loss function used for their measurement of regret must be bounded. To achieve a regret of order $O\left(1 / N_{\text {predictions }}\right)$ with probability at least $1-\delta$, they require $O\left(T^{2} \log (1 / \delta)\right)$ observations of the demonstrator.

There is a great deal of empirical study of imitation learning, given the practical applications, which Hussein et al. (2017) review. We call a few specific experiments to the reader's attention, since they resemble our work in taking an active approach to querying, with an eye to risk aversion, not just label efficiency; they find it works. First, D. S. Brown et al. (2018); D. Brown et al. (2020) consider a context where the imitator can, at any time, ask the demonstrator how it would act in any of finitely many states. These imitators focus on states that they assign higher value at risk. Those papers and the following all show strong label efficiency alongside limited loss. J. Zhang \& Cho (2017) assume some method of predicting the error of an imitator in the process of learning, and they query for help when it is above some threshold. Otherwise, their imitator follows Ross et al.'s (2011) construction. In their paper, the function that predicts the imitator's error is learned from hand-picked features of a dataset. Menda et al. (2019) query much more extensively, but like J. Zhang \& Cho (2017), they don't always act on the demonstrator's suggestion, in order to sample a more diverse set of states. Unlike J. Zhang \& Cho (2017), they do act on it when the imitator's action deviates enough from the demonstrator's (given some hand-designed distance metric over the action space). They also defer to the demonstrator when there is sufficient disagreement among an ensemble of imitators. They find their imitator is more
robust. Hoque et al. (2021) note that in many contexts, it is more convenient for the demonstrator to be queried a few times successively, rather than spread out over a long time. They modify J. Zhang \& Cho's (2017) approach: the imitator starts querying when the estimated error exceeds the same threshold, but it continues querying until it returns below a lower threshold. At the cost of more total queries, it requires fewer query-periods. Like the formal work, all these experiments regard environments that restart.

Adjacent to pure imitation learning (trying to pick the same actions as a demonstrator would), there is also work on trying to act in pursuit of the same goals as a demonstrator (which must be inferred), or matching only some outcomes of the demonstrator policy, like the expectation of some given set of features. For a review of some work in this area, see Adams et al. (2022).

### 4.3 Preliminaries

Let $a_{t} \in \mathcal{A}$ and $o_{t} \in \mathcal{O}$ be the action and observation at timestep $t \in \mathbb{N}$. Let $q_{t} \in$ $\{0,1\}$ denote whether the imitator $\left(q_{t}=0\right)$ or demonstrator $\left(q_{t}=1\right)$ selects $a_{t}$. Let $\mathcal{H}=\{0,1\} \times \mathcal{A} \times \mathcal{O}$, and let $h_{t}=\left(q_{t}, a_{t}, o_{t}\right) \in \mathcal{H}$. Let $h_{<t}=\left(h_{0}, h_{1}, \ldots, h_{t-1}\right)$. $\mathcal{X}^{n}=\times_{i=1}^{n} \mathcal{X}$ denotes the set of $n$-tuples of elements of $\mathcal{X}$, and $\mathcal{X}^{*}=\bigcup_{n=0}^{\infty} \mathcal{X}^{n}$ is the Kleene-star operator, which denotes all tuples of elements of $\mathcal{X}$.

Let $\pi: \mathcal{H}^{*} \rightsquigarrow\{0,1\} \times \mathcal{A}$, and $\rightsquigarrow$ denotes that $\pi$ gives a distribution over $\{0,1\} \times \mathcal{A} . \epsilon$ will denote the empty string; it is the element of $\mathcal{H}^{0} . \pi$ is called a policy, and will typically be written $\pi\left(q_{t} a_{t} \mid h_{<t}\right) . \pi\left(a_{t} \mid h_{<t}\right)$ denotes the marginal distribution over the action. Let $\mu: \mathcal{H}^{*} \times\{0,1\} \times \mathcal{A} \rightsquigarrow \mathcal{O}$. $\mu$ is called the environment, and will typically be written $\mu\left(o_{t} \mid h_{<t} q_{t} a_{t}\right)$. Note from this construction that an environment and a policy may qualitatively change over time-instead of being stationary with respect to the latest timestep, they can depend on the whole
history.
Much formal work in imitation learning and reinforcement learning involves defining environments in terms of their Markov states and how one transitions through them. The defining property of a state is that that future is independent of the past conditioned on the state. For those more comfortable in that framework, our state space here is $\mathcal{H}^{*}$, so the Markov property is trivial: the state is the whole history, so indeed, the future is independent of the history, when conditioned on the history. The point of the Markov Decision Process formalism is that when the state space is finite (or compact, with relevant functions of it being continuous), more tractable inference algorithms become available, but we do not assume finiteness or any structure in the state space. For finite histories denoted $h_{<t}$, the reader could mentally substitute $s_{t}$, this being the state at time $t$, but the infinite history $h_{<\infty}$, which appears in some proofs, has no standard notational analog.

Speaking of which, let $\mathcal{H}^{\infty}$ be the set of infinite strings of elements of $\mathcal{H}$. Let $\mathrm{P}_{\mu}^{\pi}$ be the probability measure over $\mathcal{H}^{\infty}$ where query records and actions are sampled from $\pi$, and observations are sampled from $\mu$. The event space is the standard sigma algebra over cylinder sets $\sigma\left(\left\{\left\{h_{<t} h_{t: \infty}: h_{t: \infty} \in \mathcal{H}^{\infty}\right\}: h_{<t} \in \mathcal{H}^{*}\right\}\right)$. In a stochastic process, a cylinder set is the set of all possible futures given a particular past.

Let $\Pi$ be a finite or countable set of policies, and for $\pi \in \Pi$, let $w(\pi)>0$ be a prior weight assigned to $\pi$, such that $\sum_{\pi \in \Pi} w(\pi)=1$. This represents the imitator's initial belief distribution over the demonstrator's policy. For convenience, let $\Pi$ only contain policies which assign zero probability to $q_{t}=0$, since demonstrator models may as well be convinced that the demonstrator is picking the action.

Example 1 ((Linear-Time) Computable Policies). The requirement that $\Pi$ be countable is not restrictive in theory. Suppose $\Pi$ is the set of programs that compute a
policy (in linear time). These can be easily enumerated, and the prior $w$ can be set $\propto 2^{- \text {program length }}$ (Kraft, 1949; Hutter, 2005).

Given the near absence of constraints, the choice of model class might pique philosophical interest. There are multiple logics with differing powers that we could plausibly use to represent programs, including "programs" higher in the arithmetic hierarchy. In general, the choice of programming language would change programs' relative length, and there are no clear desiderata when choosing a language. So Example 1 does not appear to offer an approach to solving the Problem of Priors (Talbott, 2016). The option to restrict to linear-time programs is a marginally more practical possibility that might escape most philosophical discussions.

### 4.4 Imitation

Let $w\left(\pi \mid h_{<t}\right)$ be the posterior weight after observing $h_{<t}$ that demonstratorchosen actions were sampled from $\pi$. That is,

$$
\begin{equation*}
w\left(\pi \mid h_{<t}\right): \propto w(\pi) \prod_{k<t: q_{k}=1} \pi\left(q_{k} a_{k} \mid h_{<k}\right) \tag{4.1}
\end{equation*}
$$

normalized such that $\sum_{\pi \in \Pi} w\left(\pi \mid h_{<t}\right)=1$. Ranking the policies by posterior weight, let $\pi_{n}^{h<t}$ be the one with the $n^{\text {th }}$ largest posterior weight $w\left(\pi \mid h_{<t}\right)$, breaking ties arbitrarily. Now let $\Pi_{h_{<t}}^{\alpha}$ be the set of policies with posterior weights at least $\alpha$ times the sum of the posterior weights of policies that are at least as likely as it; that is,

$$
\begin{equation*}
\Pi_{h_{<t}}^{\alpha}:=\left\{\pi_{n}^{h<t} \in \Pi: w\left(\pi_{n}^{h<t} \mid h_{<t}\right) \geq \alpha \sum_{m \leq n} w\left(\pi_{m}^{h_{<t}} \mid h_{<t}\right)\right\} \tag{4.2}
\end{equation*}
$$

This is the set of policies the imitator takes seriously. The imitator is designed to be robust to policies in this set, so smaller $\alpha$ will make it more robust. Let $\pi^{d}$ denote the demonstrator's policy, defined such that $\pi^{d}\left(q_{t}=1 \mid h_{<t}\right)=1$ for all values of $h_{<t}$. As later results suggest, $\alpha$ should be set a few orders of magnitude below $w\left(\pi^{d}\right)$; since $\pi^{d}$ is probably unknown to the programmers, or else there would be no need for imitation learning, $w\left(\pi^{d}\right)$ will have to be estimated. The imitator's policy $\pi_{\alpha}^{i}$ is defined in the next two equations:

$$
\begin{equation*}
\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right):=\min _{\pi^{\prime} \in \Pi_{h_{<t}}^{\alpha}} \pi^{\prime}\left(1, a \mid h_{<t}\right) \tag{4.3}
\end{equation*}
$$

The 0 on the l.h.s. means the imitator is picking the action itself instead of deferring to the demonstrator, and the 1 on the r.h.s. means this is the probability of the demonstrator model $\pi^{\prime}$ picking that same action.

The imitator uses the leftover probability to query. Let $\theta_{q}\left(h_{<t}\right):=1-\sum_{a \in \mathcal{A}} \pi_{\alpha}^{i}(0, a \mid$ $\left.h_{<t}\right) . \theta_{q}$ is the probability with which the imitator queries the demonstrator to have it pick the action. Thus,

$$
\begin{equation*}
\pi_{\alpha}^{i}\left(1, a \mid h_{<t}\right):=\theta_{q}\left(h_{<t}\right) \pi^{d}\left(1, a \mid h_{<t}\right) \tag{4.4}
\end{equation*}
$$

One can see that $q_{t}$ records whether the demonstrator was involved in selecting the action. Using the model class and prior from Example 1, the time-complexity constraint makes $\pi_{\alpha}^{i}$ computable.

Conservatism with respect to probability estimates is a core technical innovation of our work. Taking the minimum over a set of models with high posterior weights is an approach to conservatism inspired by M. K. Cohen \& Hutter's (2020) pessimistic agent. The pessimistic agent, unlike ours, is a reinforcement learner, but it is also designed to keep certain (risky) events unlikely. By underestimating
probabilities, the imitator only acts if it is sure the demonstrator might act that way.

We will also consider hypothetical imitator policies if the demonstrator policy were something else; for an arbitrary demonstrator policy $\pi$, let $\hat{\pi}_{\alpha}$ denote the corresponding imitator policy, so $\pi_{\alpha}^{i}=\left(\hat{\pi^{d}}\right)_{\alpha}$. This paper will investigate the probability distribution $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$ and compare it to $\mathrm{P}_{\mu}^{\pi^{d}}$.

### 4.5 Toy Example

We now walk through a toy example, in which our imitation learner has about a half-million demonstrator models in its model class $\Pi$. We begin by defining $\Pi$. The action space $\mathcal{A}$ of the demonstrator is null $\cup\{0,1\}^{4}$. The observation space $\mathcal{O}$ is $\{" ", 1,2,3\}$. ("" is the empty string). A demonstrator model $\pi \in \Pi$ defined by is a 12 -tuple of the elements $\{1 / 3,2 / 3,1\}$. When the latest observation is 1,2 , or 3 , let $x$ be the $1^{\text {st }}-4^{\text {th }}, 5^{\text {th }}-8^{\text {th }}$, or $9^{\text {th }}-12^{\text {th }}$ elements of 12 -tuple. Then, the demonstrator model outputs four bits that are Bernoulli distributed according to each of the four elements of $x$. All demonstrator models output null when the latest observation is "". The true demonstrator also takes the form of such a demonstrator model. Each observation is randomly sampled; it is 1 with probability $1 / 4$, 2 with probability $1 / 16,3$ with probability $1 / 64$, and otherwise "".

Let's give some flavor to this example. The demonstrator does client relations for a high-end travel agency with very fussy clients. The demonstrator gets a feel for her clients, and for any given night that a client needs a restaurant recommendation, the demonstrator sends a Boolean 4-tuple to the restaurant team, who identifies a suitable restaurant. The observation tells the demonstrator which of the three clients needs a recommendation, if any. The first bit of the Boolean 4-
tuple tells the restaurant team whether the restaurant should have lots of vegetarian options, the second bit: should it have a Michelin star, the third: should it have unfamiliar local specialties, and the fourth: should it be Instagrammable. Why is the demonstrator stochastic? Many clients want a variety of styles of restaurants from night to night. The demonstrator couldn't write down the exact probabilities that she is using to generate these Boolean vectors; she goes off of intuition. If we run an imitator that only sometimes asks the demonstrator for help, we can free up some of the demonstrator's time.

Unfortunately, in this toy environment, the fussy clients sometimes quit. Each client has a 4-tuple of probabilities that they would like their Boolean vector sampled from (conveniently in $\{1 / 3,2 / 3,1\}^{4}$ ). If it becomes clear that this is not how their Boolean vectors are being sampled, they quit. ("Becoming clear" is operationalized as follows: $H_{1}$ is the hypothesis that their restaurant recommendations are being sampled correctly; $H_{2}$ is the hypothesis that some other 4-tuple in $\{1 / 3,2 / 3,1\}^{4}$ is producing their restaurant recommendations. If, given the set of all restaurant recommendations they have gotten, the likelihood ratio of $H_{2}$ exceeds 100 , the client quits. Note that this happens if an element is ever False when it was supposed to be True with probability 1 ; some clients demand Michelin stars.) When recommendations are made by the demonstrator, who always correctly intuits the client's desired distribution of restaurants, clients hardly ever quit. We would like clients to hardly ever quit even when the imitator frequently takes over.

For an imitator with $\alpha=1 \mathrm{e}-14$, Figure 4.1 shows how often it has to query the demonstrator to pick the restaurant features. Recommendations are random, and this is only one run. Running it with 20 different random seeds, the number of queries required is $486.75 \pm 52.63$ (out of $2^{15}$ timesteps), and no client ever quit. Returning to run depicted in Figure 4.1, Table 4.1 works through an example of


Figure 4.1: Timesteps when the imitator queries. $2^{15}$ timesteps are shown, with black representing a query, and green representing the imitator acting unassisted. Pixels are to be read like text, left to right, top to bottom. In the accompanying code, a random seed of 0 is used to generate this image.
the posterior and the imitator's behavior. The code for this toy example can be found at https://tinyurl.com/imitation-toy-example.

### 4.6 Results

For the whole of the paper, we assume:
Assumption 1 (Realizability). $\pi^{d} \in \Pi$.
That is, the imitator can conceive of the demonstrator. There may be some interesting results in the setting of approximate realizability, where $\exists \pi \in \Pi$ such that $\pi \approx_{\varepsilon} \pi^{d}$ in some sense, but that is out of our scope here.

We now state and discuss our key results before turning to selected proofs. Our first is that the imitator's query probability converges to 0 in mean cube. This result

- renders its resemblance to the demonstrator non-trivial, since always query-

| $1 / 3$ | $2 / 3$ | 1 |
| :---: | :---: | :---: |
| $\mathbf{- 0 . 0 0 0 0}$ | -44.0000 | $-\inf$ |
| -50.0000 | $\mathbf{- 0 . 0 0 0 0}$ | $-\inf$ |
| $\mathbf{0 . 0 0 0 0}$ | -78.0000 | $-\inf$ |
| -46.0000 | $\mathbf{- 0 . 0 0 0 0}$ | $-\inf$ |
|  |  |  |
| -18.0000 | $\mathbf{- 0 . 0 0 0 0}$ | $-\inf$ |
| -69.7384 | -25.7384 | $\mathbf{- 0 . 0 0 0 0}$ |
| $\mathbf{- 0 . 0 0 0 0}$ | -22.0000 | $-\inf$ |
| -69.7384 | -25.7384 | $\mathbf{- 0 . 0 0 0 0}$ |
|  |  |  |
| $\mathbf{- 0 . 3 2 1 9}$ | -2.3219 | $-\inf$ |
| $\mathbf{- 0 . 0 0 5 6}$ | -8.0056 | - -inf |
| $\mathbf{- 0 . 0 0 0 0}$ | -16.0000 | $-\inf$ |
| $\mathbf{- 2 8 . 5 3 0 3}$ | -10.5303 | $\mathbf{- 0 . 0 0 1 0}$ |


| $p([$ False, True, False, True $] \mid$ client 2$)$ | Model |
| :---: | :---: |
| 0.11111 | $(\ldots, 2 / 3,1,2 / 3,1, \ldots)$ |
| 0.14815 | $(\ldots, 2 / 3,1 / 3,1 / 3,1, \ldots)$ |
| 0.22222 | $(\ldots, 2 / 3,1,1 / 3,1, \ldots)$ |
| 0.29630 | $(\ldots, 1 / 3,2 / 3,1 / 3,1, \ldots)$ |
| 0.44444 | $(\ldots, 1 / 3,1,1 / 3,1, \ldots)$ |

Table 4.1: Top: $\log _{2}$ posterior at timestep 1000 for the run depicted in Figure 4.1. The posterior decomposes into posterior probabilities for each of 12 features. Each block is a client, each row is a feature, and each entry is the log posterior probability that the demonstrator picks True for that feature with probability $1 / 3,2 / 3$, or 1 , respectively. To get the posterior for a whole demonstrator model, as in Equation 4.1, add the independent posteriors for each element in the 12tuple of the demonstrator model. The posterior weight on the truth is in bold for each feature; that is, the true demonstrator for this run is $(1 / 3,2 / 3,1 / 3,2 / 3$, $2 / 3,1,1 / 3,1,1 / 3,1 / 3,1 / 3,1$ ). Bottom: At timestep 1000, with $\alpha=1 \mathrm{e}-14$, we have many top models, as defined in Equation 4.2. The first column is a list of probabilities that different top models assign to the outcome [False True False True] for client 2 . The second column contains examples of top models that assign those probabilities to the outcome [False True False True] for client 2, with the true model in bold. Recall a demonstrator model is defined by a 12-tuple, but the only relevant elements for client 2 are 5-8. All these models have posterior weight large enough to make it into the top set. Thus, the probability the imitator picks [False True False True] for client 2 is 0.11111 , the minimum probability shown, as per Equation 4.3.
ing would yield perfect correspondence,

- is desirable in its own right if demonstrator access is a limited resource,
- and is instrumental in proving the remaining results, since low query probability implies little model disagreement.

Theorem 4 (Limited Querying).

$$
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)^{3}\right] \leq|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)
$$

The in mean cube bound allows infinite querying, but it diminishes in frequency, or else the expectation of an infinite sum of cubed probabilities would not be finite. Since we query under uncertainty, both querying and uncertainty diminish in tandem; this is a theme for active learners in general. Error bounds in Bayesian prediction and MAP prediction tend to be $\Theta\left(\log \left(w(\text { truth })^{-1}\right)\right)$ and $\Theta\left(w(\text { truth })^{-1}\right)$ respectively, so theoretically, our case resembles the MAP one. The cubic dependence on $\alpha$ is unfortunate, and subsequent results inherit them; the only path we found to proving a bound was fairly circuitous, and we are unsure whether this dependence can be improved.

Our remaining results show that the imitator resembles the demonstrator on one condition: $\pi^{d} \in \Pi_{h_{<t}}^{\alpha}$. Recall that $\Pi_{h_{<t}}^{\alpha}$ is a set of top demonstrator models that the imitator takes seriously, and $\pi^{d}$ is the true demonstrator model. Low model disagreement implies high accuracy when the truth is one of those models, and recall that our querying regime promises low model disagreement within finite time.

Fortunately, this condition has high probability for $\alpha \ll w\left(\pi^{d}\right)$.
Theorem 5 (Top Models Contain Truth). $\left.\mathrm{P}_{\mu^{\pi_{\alpha}^{i}}}^{i^{i}} \forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right) \geq 1-\alpha w\left(\pi^{d}\right)^{-1}$

Let $E$ be the event $\forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}$, so the true demonstrator policy is always in the top set. The high probability of $E$ is mainly of interest in the context of subsequent results that depend on it. For instance, conditioned on $E$, the imitator, when picking its own actions, converges to the demonstrator in mean cube.

Theorem 6 (Predictive Convergence). For $\alpha<w\left(\pi^{d}\right)$,

$$
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty}\left(\sum_{a \in \mathcal{A}}\left|\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)-\pi^{d}\left(1, a \mid h_{<t}\right)\right|\right)^{3} \mid E\right] \leq \frac{|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)}{1-\alpha w\left(\pi^{d}\right)^{-1}}
$$

This theorem finally justifies our calling $\pi_{\alpha}^{i}$ an "imitator", since the policy converges to that of the demonstrator. Existing literature on imitation learning does little to suggest that imitators exist in non-restarting environments. This result shows that they do, at least in a high-probability sense. Note that the denominator is the probability of $E$, which will be nearly 1 for appropriate choice of $\alpha$. The requirement that $\alpha<w\left(\pi^{d}\right)$ has important consequence: when $\alpha$ is set appropriately, the bounds in this theorem and Theorem 4 are effectively quartic in $w\left(\pi^{d}\right)^{-1}$. We do not know if a better rate is possible under additional assumptions. It is even possible that stronger results are available without additional assumptions, and we simply failed to identify them. We think this is a ripe area for research.

We argue informally that this disappointing dependence can be mitigated in some circumstances. By pre-training with $N$ consecutive demonstrator queries and calling the posterior at that point the new "prior" for the purposes of our analysis, the "prior" on $w\left(\pi^{d}\right)$ could usually be made quite large, unless most demonstrator models behave extremely similarly for the first $N$ steps. Consider an extreme case: many models of comparable weight almost agree with the true model, except one disagrees at $t=1$, one at $t=2$, etc. In this case, the posterior on the truth increases very slightly every step, as models are excluded one by one. If, on the
other hand, half of demonstrator models confidently predict one action, and half confidently predict another, the posterior on the truth will likely nearly double in one step. So to the extent that a large fraction of models in $\Pi$ disagree with $\pi^{d}$ within the first $N$ steps, the posterior on the truth would increase exponentially following pre-training. That said, the quartic dependence on $w\left(\pi^{d}\right)^{-1}$ in the worst case is a weakness of our approach.

Any pair of these first three results would be uninteresting on their own, but jointly, they show that with high probability, the imitator converges to the demonstrator with limited querying.

Our stronger results below apply when the environment and demonstrator policy do not depend on the query record. This means that whatever action is taken, the effect does not depend on whether the imitator chose it or the demonstrator did. We would like events to unfold similarly when we replace the demonstrator with the imitator, but this is impossible if the environment discriminates between them. Indeed, if the environment treats identical actions differently depending on whether they were selected by imitator or demonstrator, it's unclear what imitation accomplishes. We define fairness formally in Section 4.9.

In a fair setting, we bound the KL divergence between $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$ and $\mathrm{P}_{\mu}^{\pi^{d}}$, the first meaning that actions are picked according to our imitation policy, and the second meaning that all actions are picked by the demonstrator. The objective of imitation is most easily characterized as outputting demonstrator-like actions, but the purpose of imitation learning is for events to unfold similarly. Small errors in the limit do not guarantee that property; this result is only possible with small errors for the imitator's whole lifetime.

Theorem 7 (KL Bound). Suppose that $\mu$ and $\pi^{d}$ are fair, and $\alpha<w\left(\pi^{d}\right)$. Letting the two probability measures below be restricted to $(\mathcal{A} \times \mathcal{O})^{t}$ (that is, marginal-
izing over the query record, and considering only the first timesteps),

$$
\mathrm{KL}_{t}\left(\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(\cdot \mid E) \| \mathrm{P}_{\mu}^{\pi^{d}}(\cdot \mid E)\right) \leq \frac{\alpha^{-1}|\mathcal{A}|^{1 / 3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)^{1 / 3}}{\left(1-\alpha / w\left(\pi^{d}\right)\right)^{2}} t^{2 / 3}-\log \left(1-\alpha / w\left(\pi^{d}\right)\right)
$$

Notably, $\mathrm{KL}_{t} / t \rightarrow 0$ in the limit. The direction of the divergence resembles the variational objective (with the ground truth on the right). Thus, there may be some events that only the demonstrator would cause, but no events that only the imitator would. This consequence is made explicit in our final result.

We construct an upper bound for the probability of an event given the probability of the event if the demonstrator were acting the whole time. This bound is mainly of interest for "bad" events.

Theorem 8 (Preserving Unlikeliness). Fix t. Let $B \subset(\mathcal{A} \times \mathcal{O})^{t}$ be a (bad) event, and extending $B$ to the outcome space $(\{0,1\} \times \mathcal{A} \times \mathcal{O})^{t}=\mathcal{H}^{t}$, let $D=B \cap E$. Then, for fair $\mu$ and $\pi^{d}$,

$$
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D) \leq \frac{t^{2} s_{\alpha}}{\left(\log \frac{t^{2} s_{\alpha}}{27 \mathrm{P}_{\mu}^{\pi^{d}}(B)}-3 \log \log \left(1+\frac{t^{2 / 3} s_{\alpha}^{1 / 3}}{3 \mathrm{P}_{\mu}^{\pi^{d}(B)^{1 / 3}}}\right)\right)^{3}}
$$

where $s_{\alpha}=|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)$.
That is, as $\mathrm{P}_{\mu}^{\pi^{d}}(B)^{-1} \rightarrow \infty, \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D)^{-1} \rightarrow \infty$ at least polylogarithmicly. If an event would have been extremely unlikely under the demonstrator's policy, a similar event is unlikely when running the imitator.

Whereas existing work on imitation learners attempts to be robust to a bounded loss function, our Preserving Unlikeliness Theorem is relevant even in the absence of a uniform bound on badness. In the real world, to quote Theon Greyjoy, "It can always be worse". But some bounds on badness are possible: we tolerate one-in-
ten-chance events; they happen, and we get on with it. One-in-a-hundred-chance events can be meaningfully worse. But in a world largely governed by humans, we keep most truly devastating events below even a $1 \%$ chance. It's hard to apply similar bounds to the badness of one-in-a-billion-chance events, and in general, as the probability gets smaller, a loss function should countenance steadily larger losses. When an event goes from a $1 \%$ to a $2 \%$ chance, we should be much less concerned than if it went from $10^{-9}$ to $1 \%$. In the extreme, if an event has probability 0 under a demonstrator's policy, there might be an arbitrarily good reason for that. Whereas the bounded loss functions of all existing work ignore this effect, our Theorem 8 does not.

The main weaknesses of our results are what they require: a model class that includes the truth and a good choice of $\alpha$. Setting $\alpha$ well requires estimating $w\left(\pi^{d}\right)$, something we cannot offer general guidance on; it would depend entirely on the exact nature of the prior. And realistically, in many contexts, the realizability assumption is infeasible. There will always be mismatch between a computational model of a demonstrator and the true demonstrator. We hope this paper opens the door for other research into relaxing the realizability assumption. Plausibly, if the best approximation in $\Pi$ of $\pi^{d}$ produces certain bad events with low probability, then the imitator will too.

### 4.7 Roadmap and Notation for the Proof of Theorem 4

Much of the work of this paper is to prove Theorem 4. In this section, we state a theorem on which it depends, and we introduce the mathematical objects required to prove it.

The imitator queries when the top few demonstrator models disagree, so we bound the errors that those models can make over the agent's lifetime. We first must establish a finite bound on the errors of such models in ordinary Bayesian sequence prediction. We define that here.

Let $\mathcal{X}$ be an arbitrary finite alphabet. Let $\nu$ be a probability measure over $\mathcal{X}^{\infty}$ with the event space generated by the cylinder sets $\left\{\left\{x_{<t} x_{t: \infty} \mid x_{t: \infty} \in \mathcal{X}^{\infty}\right\} \mid\right.$ $\left.x_{<t} \in \mathcal{X}^{*}\right\}$. Let $\mathcal{M}$ be a countable set of such probability measures, and let $w(\nu)$ be a prior weight over these measures such that $\sum_{\nu \in \mathcal{M}} w(\nu)=1$. Let $x_{<t} \in \mathcal{X}^{t}$, let $\nu\left(x_{<t}\right)$ denote the probability that the infinite sequence begins with $x_{<t}$, and let $\nu\left(x \mid x_{<t}\right)=\nu\left(x_{<t} x\right) / \nu\left(x_{<t}\right)$. Let $\mu \in \mathcal{M}$ be a the "true" measure; that is, in formal results, we will let $x_{<\infty}$ be sampled from $\mu$.

Let $\nu_{n}^{x_{<t}}$ be the measure with the $n^{\text {th }}$ largest posterior weight after observing $x_{<t}$; that is, order $\mathcal{M}$ to be non-increasing in $w(\nu) \nu\left(x_{<t}\right)$, breaking ties arbitrarily, and take the $n^{\text {th }}$. (Ties between any pair should broken consistently for different $t$ ). Let the posterior $w\left(\nu \mid x_{<t}\right): \propto w(\nu) \nu\left(x_{<t}\right)$, normalized to sum to 1 . Let $\mathcal{M}_{n}^{x<t}$ be the set of the top $n$ measures, and let $w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)=\sum_{m \leq n} w\left(\nu \mid x_{<t}\right)$.

Recall a model belongs to the imitator's top set if its posterior weight is at least $\alpha$ times the sum of the posterior weights of the models that are at least as good. Thus, we define

$$
\begin{equation*}
\phi_{n}^{x_{<t}}:=\frac{w\left(\nu_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x t} \mid x_{<t}\right)} \tag{4.5}
\end{equation*}
$$

So if $\phi_{n}^{x<t} \leq \alpha$, then $\nu_{n}^{x<t}$ can be considered a "top model" in the same sense that is relevant to our imitation learner.

Our key result on which Theorem 4 is based shows that taking the minimum over predictions in the top measures converges to the truth, and the "missing probability" converges to 0 .

Theorem 9 (Top Model Convergence).

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x<t}\left(x \mid x_{<t}\right)\right]^{2} \leq \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)  \tag{i}\\
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty}\left[1-\sum_{x \in \mathcal{X}} \min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \leq|\mathcal{X}| \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)
\end{align*}
$$

This is perfectly analogous to the way the imitator predicts actions: taking the minimum over the top models for which $\phi_{n}^{x<t}>\alpha$. The difference is that in this sequence prediction setting, all observations are informative about the true measure, whereas the imitator rarely sees the demonstrator act.

To prove Theorem 9, we show that a posterior-weighted mixture over $\mathcal{M}_{n}^{x<t}$ converges to the truth, and if $\phi_{n}^{x<t}>\alpha$, then each constituent must as well. This posterior-weighted mixture is called $\rho_{n}^{\text {stat }}$. We define it here alongside other estimators that will be used in the proof of $\rho_{n}^{\text {stat }}$, convergence. First,

$$
\begin{equation*}
\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right):=\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t} x\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)} \tag{4.6}
\end{equation*}
$$

$\rho_{n}^{\text {stat }}$ resembles a maximum a posteriori estimate, but instead mixes over the top few. We call it a satis magnum a posteriori estimate (SMAP). We will show $\rho_{n}^{\text {stat }}$ converges to $\rho_{n}$, which converges to $\rho_{n}^{\text {norm }}$, which converges to $\mu$. $\rho_{n}$ and $\rho_{n}^{\text {norm }}$ are alternative SMAP estimators.
$\rho_{n}$ is not a measure, as the numerator below sums over a different set than the denominator. It sums over the top measures after observing $x$ :

$$
\begin{equation*}
\rho_{n}\left(x \mid x_{<t}\right):=\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t x}} w(\nu) \nu\left(x_{<t} x\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)} \tag{4.7}
\end{equation*}
$$

The definition appears more natural when considering a whole sequence:

$$
\begin{equation*}
\rho_{n}\left(x_{<t}\right)=\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right) \tag{4.8}
\end{equation*}
$$

Since $\sum_{x \in \mathcal{X}} \rho_{n}\left(x \mid x_{<t}\right)$ may not be 1 , we construct the measure $\rho_{n}^{\text {norm }}$ by normalizing:

$$
\begin{equation*}
\rho_{n}^{\text {norm }}\left(x \mid x_{<t}\right):=\frac{\rho_{n}\left(x \mid x_{<t}\right)}{\sum_{x^{\prime} \in \mathcal{X}} \rho_{n}\left(x^{\prime} \mid x_{<t}\right)}=\frac{\rho_{n}\left(x_{<t} x\right)}{\sum_{x^{\prime} \in \mathcal{X}} \rho_{n}\left(x_{<t} x^{\prime}\right)} \tag{4.9}
\end{equation*}
$$

Our $\rho_{n}, \rho_{n}^{\text {norm }}$, and $\rho_{n}^{\text {stat }}$ are closely inspired by Poland \& Hutter (2005), who constructed (in our notation) $\rho_{1}, \rho_{1}^{\text {norm }}$, and $\rho_{1}^{\text {stat }}$. Finally, we define the full Bayesmixture measure

$$
\begin{equation*}
\xi\left(x_{<t}\right):=\sum_{\nu \in \mathcal{M}} w(\nu) \nu\left(x_{<t}\right)=\rho_{\infty}^{\text {stat }}\left(x_{<t}\right)=\rho_{\infty}\left(x_{<t}\right)=\rho_{\infty}^{\text {norm }}\left(x_{<t}\right) \tag{4.10}
\end{equation*}
$$

We state those relationships without proof for the reader's interest; they are not used in our results.

### 4.8 General Sequence Prediction Results

This section organizes the proof of Theorem 9 into lemmas, some of which are proven here and some in Appendix A.6. We begin with elementary relations between $\xi, \rho_{n}, \rho_{n}^{\text {norm }}$, and $\rho_{n}^{\text {stat }}$.

$$
\begin{align*}
\xi\left(x_{<t}\right) & \geq \rho_{n}\left(x_{<t}\right)  \tag{4.11}\\
\rho_{n}\left(x_{<t}\right) & \geq w(\mu) \mu\left(x_{<t}\right)  \tag{4.12}\\
\rho_{n}\left(x \mid x_{<t}\right) & \geq \rho_{n}^{\text {norm }}\left(x \mid x_{<t}\right)  \tag{4.13}\\
\rho_{n}\left(x \mid x_{<t}\right) & \geq \rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right) \tag{4.14}
\end{align*}
$$

Inequalities 4.11 and 4.12 follow directly from Equation 4.8. Inequality 4.13 follows because

$$
\begin{align*}
\rho_{n}\left(x_{<t}\right)= & \max _{\mathcal{M}^{\prime} \subset \mathcal{M}:\left|\mathcal{M}^{\prime}\right|=i} \sum_{\nu \in \mathcal{M}^{\prime}} w(\nu) \nu\left(x_{<t}\right)=\max _{\mathcal{M}^{\prime} \subset \mathcal{M}:\left|\mathcal{M}^{\prime}\right|=i} \sum_{\nu \in \mathcal{M}^{\prime}} w(\nu) \sum_{x \in \mathcal{X}} \nu\left(x_{<t} x\right) \\
& \leq \sum_{x \in \mathcal{X}^{\prime}} \max _{\mathcal{M}^{\prime} \subset \mathcal{M}^{\prime}:\left|\mathcal{M}^{\prime}\right|=i} \sum_{\nu \in \mathcal{M}^{\prime}} w(\nu) \nu\left(x_{<t} x\right)=\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right) \tag{4.15}
\end{align*}
$$

so $\rho_{n}$ assigns too much probability mass. Inequality 4.14 follows because

$$
\begin{equation*}
\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)=\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t} x\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)} \leq \frac{\sum_{\nu \in \mathcal{M}_{n}^{x}<t^{x}} w(\nu) \nu\left(x_{<t} x\right)}{\rho_{n}\left(x_{<t}\right)}=\frac{\rho_{n}\left(x_{<t} x\right)}{\rho_{n}\left(x_{<t}\right)} \tag{4.16}
\end{equation*}
$$

which holds because $\mathcal{M}_{n}^{x<t x}$ is chosen to maximize the numerator.
Our first lemma bounds the normalizing factor for $\rho_{n}$, allowing us to show in our next lemma that it converges to both $\rho_{n}^{\text {norm }}$ and $\rho_{n}^{\text {stat }}$.

## Lemma 7.

$$
0 \leq \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \frac{\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right)}{\rho_{n}\left(x_{<t}\right)}-1 \leq w(\mu)^{-1}
$$

Proof idea $\rho_{n}$ is bounded above and below by measures, save a multiplicative constant (Inequalities 4.11 and 4.12), so $\rho_{n}$ converges to being a measure, in that $\sum_{x \in \mathcal{X}} \rho_{n}\left(x \mid x_{<t}\right) \rightarrow 1$.

Proof. All terms in the sum are non-negative, by Inequality 4.15. Recall $\epsilon$ denotes the empty string-the element of $\mathcal{X}^{0}$. Justifications of the upcoming lettered equations follow below the block.

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \frac{\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right)}{\rho_{n}\left(x_{<t}\right)}-1 \\
&= \sum_{t=0}^{N-1} \sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right) \frac{\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right)-\rho_{n}\left(x_{<t}\right)}{\rho_{n}\left(x_{<t}\right)} \\
& \stackrel{(a)}{\leq} \sum_{t=0}^{N-1} \sum_{x<t \in \mathcal{X}^{t}} w(\mu)^{-1}\left[\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right)-\rho_{n}\left(x_{<t}\right)\right] \\
& \stackrel{(b)}{=} w(\mu)^{-1}\left[\sum_{x_{<N} \in \mathcal{X}^{N}} \rho_{n}\left(x_{<N}\right)-\rho_{n}(\epsilon)\right] \\
& \stackrel{(c)}{\leq} w(\mu)^{-1} \sum_{x_{<N} \in \mathcal{X}^{N}} \xi\left(x_{<N}\right)=w(\mu)^{-1} \tag{4.17}
\end{align*}
$$

where (a) follows from Inequality 4.12, (b) cancels terms that are added then subtracted, and $(c)$ follows from Inequality 4.11.

Recall we are trying to show $\rho_{n}^{\text {stat }} \rightarrow \rho_{n} \rightarrow \rho_{n}^{\text {norm }} \rightarrow \mu$. The following lemma gives two of those links.

## Lemma 8.

(i)

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left|\rho_{n}\left(x \mid x_{<t}\right)-\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)\right| \leq w(\mu)^{-1}
$$

(ii)

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left|\rho_{n}\left(x \mid x_{<t}\right)-\rho_{n}^{\text {norm }}\left(x \mid x_{<t}\right)\right| \leq w(\mu)^{-1}
$$

Proof.

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left|\rho_{n}\left(x \mid x_{<t}\right)-\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)\right| \stackrel{(a)}{=} \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}} \rho_{n}\left(x \mid x_{<t}\right)-\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)= \\
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \frac{\sum_{x \in \mathcal{X}} \rho_{n}\left(x_{<t} x\right)}{\rho_{n}\left(x_{<t}\right)}-1 \stackrel{(b)}{\leq} w(\mu)^{-1} \tag{4.18}
\end{align*}
$$

where ( $a$ ) follows from Inequality 4.14 and (b) follows from Lemma 7. The proof is identical for $\rho_{n}^{\text {norm }}$, except now (a) follows from Inequality 4.13.

Given Lemma 8, the final link in showing $\rho_{n}^{\text {stat }}$ converges to $\mu$ is to show that $\rho_{n}^{\text {norm }}$ does.

Lemma 9. Recalling $\nu\left(\cdot \mid x_{<t}\right)$ is a measure over $\mathcal{X}$,

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \mathrm{KL}\left(\mu\left(\cdot \mid x_{<t}\right) \| \rho_{n}^{\text {norm }}\left(\cdot \mid x_{<t}\right)\right) \leq w(\mu)^{-1}+\log w(\mu)^{-1}
$$

Proof idea The KL divergence telescopes over timesteps. The $\log w(\mu)^{-1}$ term comes from a gap between $\mu$ and $\rho_{n}$, and the $w(\mu)^{-1}$ term comes from a gap between $\rho_{n}$ and $\rho_{n}^{\text {norm }}$.

We can now show that $\rho_{n}^{\text {stat }}$ converges to $\mu$, an independently interesting and novel result in SMAP estimation.

Theorem 10 (SMAP Convergence).

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left(\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right)^{2} \leq 6 w(\mu)^{-1}+3
$$

Proof idea $\rho_{n}^{\text {stat }}$ is close to $\rho_{n}$ in an $\ell_{1}$ sense, and likewise for $\rho_{n}$ and $\rho_{n}^{\text {norm }}$, and $\rho_{n}^{\text {norm }}$ is close to $\mu$ in an $\ell_{2}$ squared sense, since $\ell_{2}^{2} \leq$ KL. Finally, for a vector $v \in[-1,1]^{n},\|v\|_{2}^{2} \leq\|v\|_{1}$, so $\ell_{1}$ proximity implies $\ell_{2}$ proximity as well.

By applying Theorem 10 to the very similar measures $\rho_{n}^{\text {stat }}$ and $\rho_{n-1}^{\text {stat }}$, whose only difference is that the former contains $\nu_{n}^{x<t}$ in its mixture, we arrive at our final result in the general sequence prediction setting.

Theorem 9 (Top Model Convergence).

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \leq \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)  \tag{i}\\
& \mathbb{E}_{\mu} \sum_{t=0}^{\infty}\left[1-\sum_{x \in \mathcal{X}} \min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \leq|\mathcal{X}| \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)
\end{align*}
$$

Proof idea $\rho_{n}^{\text {stat }}$ is a weighted average of $\nu_{m}^{x_{<t}}$ for $m \leq n$, so convergence results for $\rho_{n}^{\text {stat }}$ and $\rho_{n-1}^{\text {stat }}$ are leveraged for $\nu_{n}^{x<t}$ 's convergence. $\phi_{n}^{x<t}>\alpha$ ensures the weights in the weighted average aren't too small, and that we only need to consider the top $\lfloor 1 / \alpha\rfloor$ models.

### 4.9 Key Proofs

We now prove our bound on the query probability, we define fairness, and we prove our bound on the probabilities of bad events.

Theorem 4 (Limited Querying).

$$
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)^{3}\right] \leq|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)
$$

Proof idea The sort of model mismatch bounded by Theorem 9 (ii) is the basis for the definition of $\theta_{q}$. Theorem 9 (ii) bounds model mismatch on observed data, and data is only observed with probability $\theta_{q}$, so with an extra factor of $\theta_{q}$ on the 1.h.s., we go from an in mean square bound to a weaker in mean cube bound.

Proof. Recall the agent considers a set of possible policies $\Pi$ that includes the true demonstrator policy $\pi^{d}$, and assigns a strictly positive prior $w(\pi)$ to each policy in $\Pi$. Recall $\mathrm{P}_{\mu}^{\pi}$ is a probability measure over $(\{0,1\} \times \mathcal{A} \times \mathcal{O})^{\infty}=\mathcal{H}^{\infty}$. Now we construct a class of measures over $\mathcal{H}^{\infty}:$ let $\mathcal{M}:=\left\{\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}: \pi \in \Pi\right\}$ (see the last paragraph of Section 4.4 for the definition of $\left.\hat{\pi}_{\alpha}\right)$, and let $w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\right):=w(\pi)$. Let $w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \mid h_{<t}\right): \propto w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\right) \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(h_{<t}\right)$. It follows straightforwardly from the definitions of the posterior that $w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \mid h_{<t}\right)=w\left(\pi \mid h_{<t}\right), w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \mid h_{<t} q_{t}\right)=$ $w\left(\pi \mid h_{<t} q_{t}\right)$, and $w\left(\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \mid h_{<t} q_{t} a_{t}\right)=w\left(\pi \mid h_{<t} q_{t} a_{t}\right)$, since all measures in $\mathcal{M}$ assign the probabilities identically to actions after $q_{t}=0$, and to observations.

Instead of saying $\mathcal{M}$ contains measures over $\mathcal{X}^{\infty}$, we generalize slightly, and say that $\mathcal{M}$ contains measures over $\times_{k=0}^{\infty} \mathcal{X}_{k}$. For $k \equiv 0 \bmod 3, \mathcal{X}_{k}=\{0,1\}$, for $k \equiv 1 \bmod 3, \mathcal{X}_{k}=\mathcal{A}$, and for $k \equiv 2 \bmod 3, \mathcal{X}_{k}=\mathcal{O}$. With $\nu_{n}^{x<k}$ and $\phi_{n}^{x<k}$ as defined before, we can apply Theorem 9 (i) to the class $\mathcal{M}$, after a trivial extension from fixed $\mathcal{X}$ to variable $\mathcal{X}_{k}$. Checking the definitions is enough to verify that $\left\{\nu_{n}^{x_{<k}}: \phi_{n}^{x_{<k}}>\alpha\right\}$ is exactly the set $\left\{\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}: \pi \in \Pi_{h_{<t}}^{\alpha}\right\}$, where $h_{j}=\left(q_{j}, a_{j}, o_{j}\right)=\left(x_{3 j}, x_{3 j+1}, x_{3 j+2}\right)$, and $t=\lfloor(k+1) / 3\rfloor$. In short, for this $\mathcal{M}$, sequence prediction errors can only come from errors predicting actions after querying, since that's when models differ, so we can use Theorem 9 to bound the latter. Recalling that $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$ is the true probability measure,

$$
\begin{align*}
& \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)=\alpha^{-3}\left(24 w\left(\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\right)^{-1}+12\right) \\
& \stackrel{(a)}{\geq} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{k=0}^{\infty} \sum_{x \in \mathcal{X}_{k}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(x \mid x_{<k}\right)-\min _{i: \phi_{n}^{x}<k>\alpha} \nu_{n}^{x_{<k}}\left(x \mid x_{<k}\right)\right]^{2} \\
& \stackrel{(b)}{=} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \sum_{q \in\{0,1\}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(q \mid h_{<t}\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(q \mid h_{<t}\right)\right]^{2}+ \\
& \sum_{a \in \mathcal{A}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(a \mid h_{<t} q_{t}\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(a \mid h_{<t} q_{t}\right)\right]^{2}+ \\
& \sum_{o \in \mathcal{O}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(o \mid h_{<t} q_{t} a_{t}\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(o \mid h_{<t} q_{t} a_{t}\right)\right]^{2} \\
& \stackrel{(c)}{=} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \sum_{a \in \mathcal{A}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(a \mid h_{<t} q_{t}\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(a \mid h_{<t} q_{t}\right)\right]^{2} \\
& =\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \sum_{q \in\{0,1\}} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(q \mid h_{<t}\right) \sum_{a \in \mathcal{A}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(a \mid h_{<t} q\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(a \mid h_{<t} q\right)\right]^{2} \\
& \stackrel{(d)}{=} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(1 \mid h_{<t}\right) \sum_{a \in \mathcal{A}}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(a \mid h_{<t} 1\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}}\left(a \mid h_{<t} 1\right)\right]^{2} \\
& \xrightarrow{(e)} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)|\mathcal{A}|\left[|\mathcal{A}|^{-1} \sum_{a \in \mathcal{A}} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(a \mid h_{<t} 1\right)-\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \mathrm{P}_{\mu}^{\hat{\mathrm{A}}_{\alpha}}\left(a \mid h_{<t} 1\right)\right]^{2} \\
& =|\mathcal{A}|^{-1} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)\left[1-\sum_{a \in \mathcal{A}} \min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \frac{\hat{\pi}_{\alpha}\left(1, a \mid h_{<t}\right)}{\hat{\pi}_{\alpha}\left(1 \mid h_{<t}\right)}\right]^{2} \\
& \stackrel{(f)}{=}|\mathcal{A}|^{-1} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)\left[1-\sum_{a \in \mathcal{A}} \min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \frac{\theta_{q}\left(h_{<t}\right) \pi\left(1, a \mid h_{<t}\right)}{\theta_{q}\left(h_{<t}\right)}\right]^{2} \\
& \stackrel{(g)}{=}|\mathcal{A}|^{-1} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)\left[\theta_{q}\left(h_{<t}\right)\right]^{2} \tag{4.19}
\end{align*}
$$

where (a) follows from Theorem 9, (b) groups triples $\left(x_{3 t}, x_{3 t+1}, x_{3 t+2}\right)$ into $h_{t}$, (c) follows because all $\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \in \mathcal{M}$ give identical conditional probabilities as $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$
on queries and observations, (d) follows because all $\mathrm{P}_{\mu}^{\hat{\pi}_{\alpha}} \in \mathcal{M}$ give identical conditional probabilities as $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$ for actions that follow $q_{t}=0,(e)$ follows from Jensen's Inequality, $(f)$ follows from the definition of $\hat{\pi}_{\alpha}$, and $(g)$ follows from the definition of $\theta_{q}\left(h_{<t}\right)$. Rearranging Inequality 4.19 gives the theorem.

Recall that Theorem 6 bounds the error between $\pi_{\alpha}^{i}$ and $\pi^{d}$, conditioned on the event $E$.

Proof idea of Theorem 6 Conditioned on $\pi^{d} \in \prod_{h_{<t}}^{\alpha}$, it follows that $\theta_{q} \geq$ the $\ell_{1}$ norm between $\pi^{d}$ and $\pi_{\alpha}^{i}$. Then we apply Theorem 4.

Our remaining theorems apply when the environment and demonstrator policy are fair. Roughly, they are fair if they do not have access to the imitator's internals.

Definition 8 (Fair). An environment $\mu: \mathcal{H}^{*} \times\{0,1\} \times \mathcal{A} \rightsquigarrow \mathcal{O}$ is fair if it does not depend on the query record; that is, $\mu\left(\cdot \mid h_{<t} q_{t} a_{t}\right)$ is not a function of $q_{k}$ for $k \leq t$. A demonstrator policy $\pi^{d}: \mathcal{H}^{*} \rightsquigarrow\{0,1\} \times \mathcal{A}$ is likewise fair if $\pi^{d}\left(\cdot \mid h_{<t}\right)$ is not a function of $q_{k}$ for $k<t$.

Theorems 7 and 8 rest on the following crux: if $\pi^{d} \in \Pi_{h_{<t}}^{\alpha}$, then $\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right) \leq$ $\pi^{d}\left(a \mid h_{<t}\right)$. Since $\pi_{\alpha}^{i}\left(1, a \mid h_{<t}\right)=\theta_{q}\left(h_{<t}\right) \pi^{d}\left(a \mid h_{<t}\right)$, we have $\pi_{\alpha}^{i}\left(a \mid h_{<t}\right) \leq$ $\left(1+\theta_{q}\left(h_{<t}\right)\right) \pi^{d}\left(a \mid h_{<t}\right)$. Thus, we have a multiplicative bound relating $\pi_{\alpha}^{i}$ and $\pi^{d}$, and it decreases to 1 .

Theorem 8 (Preserving Unlikeliness). Fix t. Let $B \subset(\mathcal{A} \times \mathcal{O})^{t}$ be a (bad) event, and extending $B$ to the outcome space $(\{0,1\} \times \mathcal{A} \times \mathcal{O})^{t}=\mathcal{H}^{t}$, let $D=B \cap E$. Then, for fair $\mu$ and $\pi^{d}$,

$$
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D) \leq \frac{t^{2} s_{\alpha}}{\left(\log \frac{t^{2} s_{\alpha}}{27 \mathrm{P}_{\mu}^{d}(B)}-3 \log \log \left(1+\frac{t^{2 / 3} 3_{\alpha}^{1 / 3}}{3 \mathrm{P}_{\mu}^{\pi_{\alpha}^{d}(B)^{1 / 3}}}\right)\right)^{3}}
$$

where $s_{\alpha}=|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)$.
Proof idea $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(B \cap E) / \mathrm{P}_{\mu}^{\pi^{d}}(B)$ increases by a factor of at most $1+\theta_{q}$ per timestep. While the expectation of $\theta_{q}^{3}$ is summable, the expectation of $\sum_{t} \theta_{q}$ grows as $O\left(t^{2}\right)$, hence that dependence in the bound. The final difficulty is that our bound on the query probability only applies in expectation, but a pathological and unlikely event $B$ could describe a case where querying is much more prolonged than expected. Thus, we do not prove a nice bound on the ratio $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(B \cap E) / \mathrm{P}_{\mu}^{\pi^{d}}(B)$. Instead, since smaller $\mathrm{P}_{\mu}^{\pi^{d}}(B)$ allows more pathology, our bound on $\mathrm{P}_{\mu}^{\tau_{\alpha}^{i}}(B \cap E)$ is only polylogarithmic in $\mathrm{P}_{\mu}^{\pi^{d}}(B)$.

Proof. If $\pi^{d} \in \Pi_{h_{<t}}^{\alpha}$, then $\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right) \leq \pi^{d}\left(a \mid h_{<t}\right)$, and of course $\pi_{\alpha}^{i}(1, a \mid$ $\left.h_{<t}\right)=\theta_{q}\left(h_{<t}\right) \pi^{d}\left(a \mid h_{<t}\right)$, so

$$
\begin{equation*}
\pi_{\alpha}^{i}\left(a \mid h_{<t}\right) \leq\left(1+\theta_{q}\left(h_{<t}\right)\right) \pi^{d}\left(a \mid h_{<t}\right) \tag{4.20}
\end{equation*}
$$

Thus, for fair $\mu$ and $\pi^{d}$, for $h_{<t} \in E$,

$$
\begin{equation*}
\frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}^{\}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}\right)} \leq \prod_{k=0}^{t-1}\left[1+\theta_{q}\left(h_{<k}\right)\right] \tag{4.21}
\end{equation*}
$$

It follows from Theorem 4 that

$$
\begin{equation*}
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{k=0}^{t-1} \theta_{q}\left(h_{<k}\right)^{3} \mid D\right] \leq \frac{s_{\alpha}}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D)} \tag{4.22}
\end{equation*}
$$

By the same derivation as in Inequality A.47, we can thus bound the sum

$$
\begin{equation*}
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{k=0}^{t-1} \theta_{q}\left(h_{<k}\right) \mid D\right] \leq t^{2 / 3}\left(\frac{s_{\alpha}}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D)}\right)^{1 / 3} \tag{4.23}
\end{equation*}
$$

Now, applying Inequality 4.20 repeatedly,

$$
\begin{aligned}
& \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\prod_{k=0}^{t-1}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1} \mid D\right] \\
& =\frac{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right) \prod_{k=0}^{t-1}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1}}{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)} \\
& \leq \frac{\sum_{h_{<t-1} \in E} \sum_{h_{t-1} \in \mathcal{H}: h_{<t} \in B} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right) \prod_{k=0}^{t-1}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1}}{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)} \\
& =\frac{\sum_{h_{<t-1} \in E}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t-1}\right) \prod_{k=0}^{t-2}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1}\right]}{\sum_{h_{<t \in D}} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)} \\
& \sum_{h_{t-1} \in \mathcal{A} \times \mathcal{O}: h_{<t} \in B} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{t-1}^{\backslash} \mid h_{<t-1}\right)\left(1+\theta_{q}\left(h_{<t-1}\right)\right)^{-1}
\end{aligned}
$$

$\stackrel{(a)}{\leq} \frac{\sum_{h_{<t-1} \in E}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t-1}\right) \prod_{k=0}^{t-2}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1}\right] \sum_{h_{t-1}^{\backslash} \in \mathcal{A} \times \mathcal{O}: h_{<t} \backslash B} \mathrm{P}_{\mu}^{\pi^{d}}\left(h_{t-1}^{\backslash} \mid h_{<t-1}\right)}{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}$
$\stackrel{(b)}{\leq} \frac{\sum_{h_{<t-2} \in E}\left[\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t-2}\right) \prod_{k=0}^{t-3}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1}\right] \sum_{h_{t-2} h_{t-1}^{\backslash} \in(\mathcal{A} \times \mathcal{O})^{2}: h_{<t}^{\backslash} \in B} \mathrm{P}_{\mu}^{\pi^{d}}\left(h_{t-2}^{\backslash} h_{t-1}^{\backslash} \mid h_{<t-2}\right)}{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}$
$\stackrel{(c)}{\leq} \frac{\sum_{h_{<t} \in B} \mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}^{\backslash}\right)}{\sum_{h_{<t} \in D} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}=\frac{\mathrm{P}_{\mu}^{\pi^{d}}(B)}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D)}$
where ( $a$ ) follows from Inequality 4.20 since $h_{<t-1} \in E$ (note the change from $\pi_{\alpha}^{i}$ to $\left.\pi^{d}\right),(b)$ iterates the previous three lines, and $(c)$ iterates the logic down to 0 .

Now we bound the expectation

$$
\begin{align*}
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\prod_{k=0}^{t-1}\left(1+\theta_{q}\left(h_{<k}\right)\right)^{-1} \mid D\right] & \stackrel{(a)}{\geq} \prod_{k=0}^{t-1}\left(1+\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\theta_{q}\left(h_{<k}\right) \mid D\right]\right)^{-1} \\
& =\exp \left(-\sum_{k=0}^{t-1} \log \left(1+\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\theta_{q}\left(h_{<k}\right) \mid D\right]\right)\right) \\
& \geq \exp \left(-\sum_{k=0}^{t-1} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\theta_{q}\left(h_{<k}\right) \mid D\right]\right) \\
& \stackrel{(b)}{\geq} e^{-t^{2 / 3} s_{\alpha}^{1 / 3} P_{\mu}^{\pi_{\alpha}^{i}}(D)^{-1 / 3}} \tag{4.25}
\end{align*}
$$

where ( $a$ ) follows from Jensen's Inequality (one can easily show the Hessian of $\prod_{i} 1 /\left(1+x_{i}\right)$ is positive semidefinite for $\left.x \succ 0\right)$, and $(b)$ follows from Inequality 4.23. Solving for $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D)$ in terms of $\mathrm{P}_{\mu}^{\pi^{d}}(B)$, we get

$$
\begin{equation*}
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D) \leq \frac{t^{2} s_{\alpha}}{27 W\left(\frac{t^{2 / 3} s_{\alpha}^{1 / 3}}{\left.3 \mathrm{P}_{\mu}^{\pi^{d}(B)^{1 / 3}}\right)^{3}}\right.} \tag{4.26}
\end{equation*}
$$

where $W$ is the Lambert- $W$ function, defined by the property $W(z) e^{W(z)}=z$. A property of the Lambert- $W$ function-that $W(z) \geq \log z-\log \log (1+z)$-yields the theorem:

$$
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(D) \leq \frac{t^{2} s_{\alpha}}{\left(\log \frac{t^{2} s_{\alpha}}{27 \mathrm{P}_{\mu}^{d^{d}}(B)}-3 \log \log \left(1+\frac{t^{2 / 3} s_{\alpha}^{1 / 3}}{3 \mathrm{P}_{\mu}^{d}(B)^{1 / 3}}\right)\right)^{3}}
$$

One can easily verify this inequality by supposing the opposite and showing that it violates Inequality 4.25 , but we omit this.

### 4.10 Conclusion

We present the first formal results for an imitation learner in a setting where the environment does not reset. We present the first formal results for an imitation learner that do not depend on a bounded loss assumption. We present the first finite error bounds for an agent acting in general environments; existing results only regard limiting behavior (although existing work considers reinforcement learning, a harder problem than imitation learning). If we would like to have an artificial agent imitate, with particular concern for keeping unlikely events unlikely, this is the first theory of how to do it.

## Acknowledgements

This work was supported by the Leverhulme Centre for the Future of Intelligence, Australian Research Council Discovery Projects DP150104590, the Oxford-Man Institute, and the Berkeley Existential Risk Initiative. Thank you to Jan Leike for encouraging us to write a paper on imitation learning.

### 4.11 End notes

Some closing comments to clarify: I mention that $\alpha$ should be set a few orders of magnitude below $w\left(\pi^{d}\right)$. This could be a key challenge of this approach, since it may be hard to estimate $w\left(\pi^{d}\right)$. One might note that in the toy example from Section 4.5, $\alpha$ is set to be very small. In practice, one might start with an extremely low value of $\alpha$ and increase over time if that seems necessary for the agent to actually act.

When I say that existing work in imitation learning makes an i.i.d. assumption
about the trajectories, I mean that they assume each trajectory is sampled from the same distribution independently. This is a common kind of assumption that appears in the supervised learning literature, but it is not the only assumption from the literature that is called an i.i.d. assumption. In much formal work, it is instead assumed that the model's error is i.i.d. across data points.

## 5| Fast Uncertainty Quantification


#### Abstract

Gaussian processes (GPs) produce good probabilistic models of functions, but most GP kernels require $O\left((n+m) n^{2}\right)$ time, where $n$ is the number of data points and $m$ the number of predictive locations. We present a new kernel that allows for Gaussian process regression in $O((n+m) \log (n+m))$ time. Our "binary tree" kernel places all data points on the leaves of a binary tree, with the kernel depending only on the depth of the deepest common ancestor. We can store the resulting kernel matrix in $O(n)$ space in $O(n \log n)$ time, as a sum of sparse rank-one matrices, and approximately invert the kernel matrix in $O(n)$ time. Sparse GP methods also offer linear run time, but they predict less well than higher dimensional kernels. On a classic suite of regression tasks, we compare our kernel against Matérn, sparse, and sparse variational kernels. The binary tree GP assigns the highest likelihood to the test data on a plurality of datasets, usually achieves lower mean squared error than the sparse methods, and often ties or beats the Matérn GP. On large datasets, the binary tree GP is fastest, and much faster than a Matérn GP.


### 5.1 Introduction

Gaussian processes (GPs) can be used to perform regression with high-quality uncertainty estimates, but they are slow. Naïvely, GP regression requires $O\left(n^{3}+\right.$ $\left.n^{2} m\right)$ computation time and $O\left(n^{2}\right)$ computation space when predicting at $m$ locations given $n$ data points (C. K. Williams \& Rasmussen, 2006). A kernel matrix of size $n \times n$ must be inverted (or Cholesky decomposed), and then $m$ matrix-vector
multiplications must be done with that inverse matrix (or $m$ linear solves with the Cholesky factors). A few methods that we will discuss later achieve $O\left(n^{2} m\right)$ time complexity (Wang et al., 2019; Y. Zhang et al., 2005).

With special kernels, GP regression can be faster and use less space. Inducing point methods, using $z$ inducing points, allow regression to be done in $O\left(z^{2}(n+\right.$ $m)$ ) time and in $O\left(z^{2}+z n\right)$ space (Quinonero-Candela \& Rasmussen, 2005; Snelson \& Ghahramani, 2005; Titsias, 2009b; Hensman et al., 2013). We will discuss the details of these inducing point kernels later, but they are kernels in their own right, not just approximations to other kernels. Unfortunately, these kernels are low dimensional (having a $z$-dimensional Hilbert space), which limits the expressivity of the GP model.

We present a new kernel, the binary tree kernel, that also allows for GP regression in $O(n+m)$ space and $O((n+m) \log (n+m))$ time (both model fitting and prediction). The time and space complexity of our method is also linear in the depth of the binary tree, which is naïvely linear in the dimension of the data, although in practice we can increase the depth sublinearly. Training some kernel parameters takes time quadratic in the depth of the tree. The dimensionality of the binary tree kernel is exponential in the depth of the tree, making it much more expressive than an inducing points kernel. Whereas for an inducing points kernel, the runtime is quadratic in the dimension of the Hilbert space, for the binary tree kernel, it is only logarithmic-an exponential speedup.

A simple depiction of our kernel is shown in Figure 5.1, which we will define precisely in Section 5.3. First, we create a procedure for placing all data points on the leaves of a binary tree. Given the binary tree, the kernel between two points depends only on the depth of the deepest common ancestor. Because very different tree structures are possible for the data, we can easily form an ensemble


Figure 5.1: A binary tree kernel with four data points. In this example, $k\left(x_{1}, x_{1}\right)=$ $1, k\left(x_{1}, x_{2}\right)=0, k\left(x_{1}, x_{3}\right)=0.8$, and $k\left(x_{1}, x_{4}\right)=0.3$.
of diverse GP regression models. Figure 5.2 depicts a schematic sample from a binary tree kernel. Note how the posterior mean is piecewise flat, but the pieces can be small.

On a standard suite of benchmark regression tasks (Wang et al., 2019), we show that our kernel usually achieves better negative log likelihood (NLL) than state-of-the-art sparse methods and conjugate-gradient-based "exact" methods, at lower computational cost in the big-data regime.

There are not many limitations to using our kernel. The main limitation is that other kernels sometimes capture the relationships in the data better. We do not have a good procedure for understanding when data has more Matérn character or more binary tree character (except through running both and comparing training NLL). But given that the binary tree kernel usually outperforms the Matérn, we'll tentatively say the best first guess is that a new dataset has more binary tree char-


Figure 5.2: A schematic diagram of a function sampled from a binary tree kernel. The function is over the interval $[0,1]$, and points on the interval are placed onto the leaves of a depth-4 binary tree according to the first 4 bits of their binary expansion. The sampled function is in black. Purple represents the sample if the tree had depth 3 , green depth 2 , orange depth 1 , and red depth 0 .
acter. One concrete limitation for some applications, like Bayesian Optimization, is that the posterior mean is piecewise-flat, so gradient-based heuristics for finding extrema would not work.

In contexts where a piecewise-flat posterior mean is suitable, we struggle to see when one would prefer a sparse or sparse variational GP to a binary tree kernel. The most thorough approach would be to run both and see which has a better training NLL, but if you had to pick one, the binary tree GP seems to be better performing and comparably fast. If minimizing mean-squared error is the objective, the Matern kernel seems to do slightly better than the binary tree. If the dataset is small, and one needs a very fast prediction, a Matérn kernel may be the best option. But otherwise, if one cares about well-calibrated predictions, these initial results we present tentatively suggest using a binary tree kernel over the widely-used Matérn kernel.

The log-linear time and linear space complexity of the binary tree GP, with performance exceeding a "normal" kernel, could profoundly expand the viability of GP regression to larger datasets.

### 5.2 Preliminaries

Our problem setting is regression. Given a function $f: \mathcal{X} \rightarrow \mathbb{R}$, for some arbitrary set $\mathcal{X}$, we would like to predict $f(x)$ for various $x \in \mathcal{X}$. What we have are observations of $f(x)$ for various (other) $x \in \mathcal{X}$. Let $X \in \mathcal{X}^{n}$ be an $n$-tuple of elements of $\mathcal{X}$, and let $y \in \mathbb{R}^{n}$ be an $n$-tuple of real numbers, such that $y_{i} \sim$ $f\left(X_{i}\right)+\mathcal{N}(0, \lambda)$, for $\lambda \in \mathbb{R}^{\geq 0} . X$ and $y$ comprise our training data.

With an $m$-tuple of test locations $X^{\prime} \in \mathcal{X}^{m}$, let $y^{\prime} \in \mathbb{R}^{m}$, with $y_{i}^{\prime}=f\left(X_{i}^{\prime}\right) . y^{\prime}$ is the ground truth for the target locations. Given training data, we would like to
produce a distribution over $\mathbb{R}$ for each target location $X_{i}^{\prime}$, such that it assigns high marginal probability to the unknown $y_{i}^{\prime}$. Alternatively, we sometimes would like to produce point estimates $\hat{y}_{i}^{\prime}$ in order to minimize the squared error $\left(\hat{y}_{i}^{\prime}-y_{i}^{\prime}\right)^{2}$.

A GP prior over functions is defined by a mean function $m: \mathcal{X} \rightarrow \mathbb{R}$, and a kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. The expected function value at a point $x$ is defined to be $m(x)$, and the covariance of the function values at two points $x_{1}$ and $x_{2}$ is defined to be $k\left(x_{1}, x_{2}\right)$. Let $K_{X X} \in \mathbb{R}^{n \times n}$ be the matrix of kernel values $\left(K_{X X}\right)_{i j}=k\left(X_{i}, X_{j}\right)$, and let $m_{X} \in \mathbb{R}^{n}$ be the vector of mean values $\left(m_{X}\right)_{i}=m\left(X_{i}\right)$. For a GP to be well-defined, the kernel must be such that $K_{X X}$ is positive semidefinite for any $X \in \mathcal{X}^{n}$. For a point $x \in \mathcal{X}$, Let $K_{X x} \in \mathbb{R}^{n}$ be the vector of kernel values: $\left(K_{X x}\right)_{i}=k\left(X_{i}, x\right)$, and let $K_{x X}=K_{X x}^{\top}$. Let $\lambda \geq 0$ be the variance of observation noise. Let $\mu_{x}$ and $\sigma_{x}^{2}$ be the mean and variance of our posterior predictive distribution at $x$. Then, with $K_{X X}^{\lambda i n v}=\left(K_{X X}+\lambda I\right)^{-1}$,

$$
\begin{align*}
\mu_{x} & :=\left(y-m_{X}\right)^{\top}\left(K_{X X}+\lambda I\right)^{-1} K_{X x}+m(x)  \tag{5.1}\\
\sigma_{x}^{2} & :=k(x, x)-K_{x X}\left(K_{X X}+\lambda I\right)^{-1} K_{X x}+\lambda \tag{5.2}
\end{align*}
$$

See C. K. Williams \& Rasmussen (2006) for a derivation. We compute Equations 5.1 and 5.2 for all $x \in X^{\prime}$.

### 5.3 Binary tree kernel

We now introduce the binary tree kernel. First, we encode our data points as binary strings. So we have $\mathcal{X}=\mathbb{B}^{q}$, where $\mathbb{B}=\{0,1\}$, and $q \in \mathbb{N}$.

If $\mathcal{X}=\mathbb{R}^{d}$, we must map $\mathbb{R}^{d} \mapsto \mathbb{B}^{q}$. First, we rescale all points (training points and test points) to lie within the box $[0,1]^{d}$. (If we have a stream of test points, and one lands outside of the box $[0,1]^{d}$, we can either set $K_{x X}$ to $\mathbf{0}$ for that point
or we rescale and retrain in $O(n \log n)$ time.) Then, for each $x \in[0,1]^{d}$, for each dimension, we take the binary expansion up to some precision $p$, and for those $d \times p$ bits, we permute them using some fixed permutation. We call this permutation the bit order, and it is the same for all $x \in[0,1]^{d}$. Note that now $q=d p$. See Figure 5.3 for an example. We optimize the bit order during training, and we can also form an ensemble of GPs using different bit orders.

For $x \in \mathbb{B}^{q}$, let $x^{\leq i}$ be the first $i$ bits of $x$. 【expression】 evaluates to 1 , if expression is true, otherwise 0 . We now define the kernel:

Definition 9 (Binary Tree Kernel). Given a weight vector $w \in \mathbb{R}^{q}$, with $w \succeq 0$ and $\|w\|_{1}=1$,

$$
k_{w}\left(x_{1}, x_{2}\right)=\sum_{i=1}^{q} w_{i}\left[\left[x_{1}^{\leq i}=x_{2}^{\leq i}\right]\right]
$$



Figure 5.3: Function from $[0,1]^{2} \rightarrow \mathbb{B}^{8}$.

So the more leading bits shared by $x_{1}$ and $x_{2}$, the larger the covariance between the function values. Consider, for example, points $x_{1}$ and $x_{4}$ from Figure 5.1, where $x_{1}$ is (left, left, right), and $x_{4}$ is (left, right, right); they share only the first leading "bit". We train the weight vector $w$ to maximize the likelihood of the training data.

Proposition 1 (Positive Semidefiniteness). For $X \in \mathcal{X}^{n}$, for $k=k_{w}, K_{X X} \succeq 0$.

Proof. Let $s \in \bigcup_{i=1}^{q} \mathbb{B}^{i}$ be a binary string, and let $|s|$ be the length of $s$. Let $X_{[s]} \in \mathbb{R}^{n}$ with $\left(X_{[s]}\right)_{j}=\left[\left[X_{j}^{\leq|s|}=s\right]\right] . X_{[s]} X_{[s]}^{\top}$ is clearly positive semidefinite. Finally, $K_{X X}=\sum_{i=1}^{q} \sum_{s \in \mathbb{B}^{i}} w_{i} X_{[s]} X_{[s]}^{\top}$, and recall $w_{i} \geq 0$, so $K_{X X} \succeq 0$.

### 5.4 Sparse rank one sum representation

In order to do GP regression in $O(n)$ space and $O(n \log n)$ time, we develop a "Sparse Rank One Sum" representation of linear operators (SROS). This was developed separately from the very similar Hierarchical matrices (Bebendorf, 2008), which we discuss below. In SROS form, linear transformation of a vector can be done in $O(n)$ time instead of $O\left(n^{2}\right)$. We will store our kernel matrix and inverse kernel matrix in SROS form. The proof of Proposition 1 exemplifies representing a matrix as the sum of sparse rank one matrices. Note that each $X_{[s]}$ is sparse-if $q$ is large, most $X_{[s]}$ 's are the zero vector.

We now show how to interpret an SROS representation of an $n \times n$ matrix. Let $[n]=\{1,2, \ldots, n\}$. For $r \in \mathbb{N}$, let $L:[r]^{n} \times[r]^{n} \times \mathbb{R}^{n} \times \mathbb{R}^{n} \rightarrow \mathbb{R}^{n \times n}$ construct a linear operator from four vectors.

Definition 10 (Linear Operator from Simple SROS Representation). Let p, $p^{\prime} \in$ $[r]^{n}$, and let $u, u^{\prime} \in \mathbb{R}^{n}$. For $l \in[r]$, let $u^{p=l} \in \mathbb{R}^{n}$ be the vector where $u_{j}^{p=l}=$ $u_{j} \llbracket p_{j}=l \rrbracket$, likewise for $u^{\prime}$ and $p^{\prime}$. Then: $L\left(p, p^{\prime}, u, u^{\prime}\right) \mapsto \sum_{i=1}^{m} u^{p=i}\left(u^{\prime}\right)^{p^{\prime}=i^{\top}}$.

We depict Definition 10 in Figure 5.4. $p$ and $p^{\prime}$ represent partitions over $n$ ele-


Figure 5.4: A matrix in standard form constructed from a matrix in SROS form. The large square depicts the matrix $L\left(p, p^{\prime}, u, u^{\prime}\right) \in \mathbb{R}^{5 \times 5}$ with elements colored by value. See $u^{p=0}$ for a color legend.
ments: all elements with the same integer value in the vector $p$ belong to the same partition. Note that $r$, the number of parts in the partition, need not exceed $n$, the number of elements being partitioned. If $p=p^{\prime}$ (which is almost always the case for us) and the elements of $p, u$, and $u^{\prime}$ were shuffled so that all elements in the same partition were next to each other, then $L\left(p, p^{\prime}, u, u^{\prime}\right)$ would be block diagonal. Note that $L\left(p, p^{\prime}, u, u^{\prime}\right)$ is not necessarily low rank. If $p$ is the finest possible partition, and $p=p^{\prime}, L\left(p, p^{\prime}, u, u^{\prime}\right)$ is diagonal. SROS matrices can be thought of as a generalization of two types of matrix that are famously amenable to fast computation: rank one matrices (all points in the same partition) and diagonal matrices (each point in its own partition).

We now extend the definition of $L$ to allow for multiple $p, p^{\prime}, u$, and $u^{\prime}$ vectors.
Definition 11 (Linear Operator from SROS Representation). Let $L:[r]^{n \times q} \times$ $[r]^{n \times q} \times \mathbb{R}^{n \times q} \times \mathbb{R}^{n \times q} \rightarrow \mathbb{R}^{n \times n}$. Let $P, P^{\prime} \in[r]^{n \times q}$, and let $U, U^{\prime} \in \mathbb{R}^{n \times q}$. Let $P_{:, i}, U_{:, i}$, etc. be the $i^{\text {th }}$ columns of the respective arrays. Then: $L\left(P, P^{\prime}, U, U^{\prime}\right) \mapsto$ $\sum_{i=1}^{q} L\left(P_{:, i}, P_{:, i}^{\prime}, U_{:, i}, U_{:, i}^{\prime}\right)$.

Algorithm 1 performs linear transformation of a vector using SROS representation in $O(n q)$ time.

We now discuss how to approximately invert a certain kind of symmetric SROS matrix, but our methods could be extended to asymmetric matrices. First, we define a partial ordering over partitions. For two partitions $p, p^{\prime}$, we say $p^{\prime} \leq p$ if $p^{\prime}$ is finer than or equal to $p$; that is, $p_{j}^{\prime}=p_{j^{\prime}}^{\prime} \Longrightarrow p_{j}=p_{j^{\prime}}$. Using that partial ordering, a symmetric SROS matrix can be approximately inverted efficiently if for all $1 \leq i, i^{\prime} \leq q, P_{:, i} \leq P_{:, i^{\prime}}$ or $P_{:, i i^{\prime}} \leq P_{:, i}$. As the reader may have recognized, our kernel matrix $K_{X X}$ can be written as an SROS matrix with this property.

We will write symmetric SROS matrices in a slightly more convenient form. All $\left(u^{\prime}\right)^{p=l}$ must be a constant times $u^{p=l}$. We will store these constants in an array

```
Algorithm 1 Linear Transformation with SROS Linear Operator. This
can be vectorized on a Graphical Processing Unit (GPU), using e.g.
torch.Tensor.index_add_ for Line 5 and non-slice indexing for Line 6
(Paszke et al., 2019). Slight restructuring allows vectorization over \([q]\) as well.
Require: \(P, P^{\prime} \in[r]^{n \times q}, U, U^{\prime} \in \mathbb{R}^{n \times q}, x \in \mathbb{R}^{n}\)
Ensure: \(y=L\left(P, P^{\prime}, U, U^{\prime}\right) x\)
    \(y \leftarrow \mathbf{0} \in \mathbb{R}^{n}\)
    for \(i \in[q]\) do \(\triangleright O(n q)\) time
        \(p, p^{\prime}, u, u^{\prime} \leftarrow P_{:, i}, P_{:, i}^{\prime}, U_{:, i}, U_{:, i}^{\prime}\)
        \(z \leftarrow \mathbf{0} \in \mathbb{R}^{r} \quad \triangleright z_{l}\) will store the dot product \(\left(\left(u^{\prime}\right)^{p^{\prime}=l}\right)^{\top} x^{p^{\prime}=l}\)
        for \(j \in[n]\) do \(z_{p_{j}^{\prime}} \leftarrow z_{p_{j}^{\prime}}+u_{j}^{\prime} x_{j} \quad \triangleright O(n)\) time
        for \(j \in[n]\) do \(y_{j} \leftarrow y_{j}+z_{p_{j}} u_{j} \quad \triangleright O(n)\) time
    return \(y\)
```

$C$. Let $L(P, C, U)$ be shorthand for $L(P, P, U, C \odot U)$, where $\odot$ denotes elementwise multiplication. For $L(P, C, U)$ to be symmetric, it must be the case that $P_{j i}=P_{j^{\prime} i} \Longrightarrow C_{j i}=C_{j^{\prime} i}$. Then, all elements of $U$ corresponding to a given $u^{p=l}$ are multiplied by the same constant. We now present an algorithm for calculating $(L(P, C, U)+\lambda I)^{-1}$, for $\lambda \neq 0$, which is an approximate inversion of $L(P, C, U)$. We have not yet analyzed numerical sensitivity for $\lambda \rightarrow 0$, but we conjecture that all floating point numbers involved need to be stored to at least $\log _{2}(1 / \lambda)$ bits. Without loss of generality, let $\lambda=1$, and note $(L(P, C, U)+\lambda I)^{-1}=$ $\lambda^{-1}\left(L\left(P, \lambda^{-1} C, U\right)+I\right)^{-1}$.

By assumption, all columns of $P$ are comparable with respect to the partial ordering above, so we can reorder the columns of $P$ such that $P_{:, i} \geq P_{:, j}$ for $i<j$. The key identity that we use to develop our fast inversion algorithm is the ShermanMorrison Formula:

$$
\begin{equation*}
\left(A+c u u^{\top}\right)^{-1}=A^{-1}-\frac{A^{-1} u u^{\top} A^{-1}}{c^{-1}+u^{\top} A^{-1} u} \tag{5.3}
\end{equation*}
$$

Starting with $A=I$, we add the sparse rank one matrices iteratively, from the
finest partition to the coarsest one, updating $A^{-1}$ as we go. We represent $(L(P, C, U)+$ $I)^{-1}$ in the form $I+L\left(P, C^{\prime}, U^{\prime}\right)$, so we write an algorithm that returns $C^{\prime}$ and $U^{\prime}$. We can also quickly calculate $\log |L(P, C, U)+I|$ at the same time, using the matrix determinant lemma: $\left|A+c u u^{\top}\right|=\left(1+c u^{\top} A^{-1} u\right)|A|$.
Theorem 11 (Fast Inversion). For $P \in[r]^{n \times q}$ and $C, U \in \mathbb{R}^{n \times q}$, if $P_{:, i} \stackrel{\text { (is coarser than) }}{\geq} P_{:, j}$ for $i<j$, then there exists $C^{\prime}, U^{\prime} \in \mathbb{R}^{n \times q}$, such that $(L(P, C, U)+I)^{-1}=$ $I+L\left(P, C^{\prime}, U^{\prime}\right)$. There exists an algorithm for computing $C^{\prime}$ and $U^{\prime}$ that takes $O\left(n q^{2}\right)$ time.

Proof. For $X \in \mathbb{R}^{n \times q}$, let $X_{:, i+1: q} \in \mathbb{R}^{n \times(q-i)}$ be columns $i+1$ through $q$ of matrix $X$ (inclusive). Let $A_{i}=I+L\left(P_{:, i+1: q}, C_{:, i+1: q}, U_{\mathrm{B}, i+1: q}\right)$, and $A_{q}=I$. Now suppose $A_{i}^{-1}$ can be written as $I+L\left(P_{:, i+1: q}, C_{:, i+1: q}^{\prime}, U_{:, i+1: q}^{\prime}\right)$ for some $C^{\prime}$ and $U^{\prime}$. For the base case of $i=q$, this holds trivially. We show it also holds for $i-1$, and we can compute $C_{:, i: q}^{\prime}, U_{:, i: q}^{\prime}$ in $O(n(q-i))$ time. Let $p=P_{:, i}, u=U_{:, i}$, and $c=C_{:, i}$. Consider $u^{p=l}$, where each element is zero unless the corresponding element of $p$ equals $l$. What do we know about the product $A_{i}^{-1} u^{p=l}$ (as seen in Equation 5.3)?

Because the columns of $P$ go from coarser partitions to finer ones, all of the vectors generating the sparse rank one components of $L\left(P_{:, i+1: q}, C_{:, i+1: q}^{\prime}, U_{:, i+1: q}^{\prime}\right)$ are from partitions that are equal to or finer than $p$. Thus, they are either zero everywhere $u^{p=l}$ is zero, or zero everywhere $u^{p=l}$ is nonzero. Vectors $v$ of the second kind can be ignored, as $c v v^{\top} u^{p=l}=0$. Thus, when multiplying $L\left(P_{:, i+1: q}, C_{:, i+1: q}^{\prime}, U_{:, i+1: q}^{\prime}\right)$ by $u^{p=l}$, the only relevant vectors are filled with zeros except where the corresponding element of $p$ equals $l$. So we can get rid of those rows of $P_{:, i+1: q}$, $C_{:, i+1: q}^{\prime}$, and $U_{:, i+1: q}^{\prime}$. Suppose there are $n_{l}$ elements of $p$ that equal $l$. Then $L\left(P_{:, i+1: q}, C_{:, i+1: q}^{\prime}, U_{:, i+1: q}^{\prime}\right) u^{p=l}$ involves $n_{l}$ rows, and can be computed in $O\left(n_{l}(q-\right.$ $i)$ ) time. Moreover, this product, which we'll call $\left(u^{\prime}\right)^{p=l}$, is only nonzero when
the corresponding element of $p$ equals $l$, so it has the same sparsity pattern as $u^{p=l}$. The other component of $A_{i}^{-1}$ is the identity matrix, and $I u^{p=l}$ clearly has the same sparsity as $u^{p=l}$. Thus, returning to Equation 5.3, when we add $u^{p=l}\left(u^{p=l}\right)^{\top}$ to $A_{i}$, we update $A_{i}^{-1}$ with an outer product of vectors whose sparsity pattern is the same as that of $u^{p=l}$.

For each $l, A_{i}^{-1}$ need not be updated with each $u^{p=l}$ one at a time. For $l \neq l^{\prime}$, $u^{p=l}$ and $u^{p=l^{\prime}}$ are nonzero at separate indices, so $u^{p=l}$ and $\left(u^{\prime}\right)^{p=l^{\prime}}$ are nonzero at separate indices, so the extra component of $A_{i}^{-1}$ that appears after the $u^{p=l^{\prime}}$ update is irrelevant to the $u^{p=l}$ update, because $\left(u^{p=l}\right)^{\top}\left(u^{\prime}\right)^{p=l^{\prime}}=0$. Since the $u^{p=l}$ update takes $O\left(n_{l}(q-i)\right)$ time, all of them together take $O\left(\sum_{l} n_{l}(q-i)\right)$ time, which equals $O(n(q-i))$ time. Calculating an element of $c^{\prime}$ only involves computing the denominator in Equation 5.3, using a matrix-vector product already computed. So we can write $C_{:, i: q}^{\prime}$ and $U_{:, i: q}^{\prime}$ by adding a preceding column to $C_{:, i+1: q}^{\prime}$ and $U_{:, i+1: q}^{\prime}$, using the same partition $p$, and it takes $O(n(q-i))$ time.

Following the induction down to $i=0$, we have $(L(P, C, U)+I)^{-1}=I+$ $L\left(P, C^{\prime}, U^{\prime}\right)$, and a total time of $O\left(n q^{2}\right)$.

Algorithm 2 also performs approximate inversion, which we prove in Appendix A.7. It differs slightly from the algorithm in the proof, but can take full advantage of a GPU speedup. In the setting where all columns of $U$ are identical, observe that in Lines 10 and 11, the same computation is repeated for all $k \in[i]$. Indeed, in this setting, this block of code can be modified to run in $O(n)$ time rather than $O(n i)$, making the whole algorithm run in $O(n q)$ time, as shown in Proposition 4 in Appendix A.7.

A Hierarchical matrix is a matrix which is either represented as a low-rank matrix or as a $2 \times 2$ block matrix of Hierarchical matrices (Bebendorf, 2008). In our SROS format, many of the sparse rank one matrices overlap, whereas in a Hierarchical

```
Algorithm 2 Inverse and determinant of \(I+\) SROS Linear Operator. Lines
5 through 11 can all be easily vectorized on a GPU. Lines 5 and 10 require
torch. Tensor.index_add_or equivalent, and lines 6 and 11 require non-
slice indexing, which are not quite as fast as some GPU operations.
Require: \(P \in[r]^{n \times q}, C, U \in \mathbb{R}^{n \times q}\)
Ensure: \(I+L\left(P, C^{\prime}, U^{\prime}\right)=(I+L(P, C, U))^{-1} ; x=\log |I+L(P, C, U)|\)
    \(x, C^{\prime}, U^{\prime} \leftarrow 0, \mathbf{0} \in \mathbb{R}^{n \times q}, U\)
    for \(i \in(q, q-1, \ldots, 1)\) do \(\quad \triangleright O\left(n q^{2}\right)\) time
        \(p, c, u, u^{\prime} \leftarrow P_{:, i}, C_{:, i}, U_{:, i}, U_{:, i}^{\prime}\)
        \(z \leftarrow \mathbf{0} \in \mathbb{R}^{r} \quad \triangleright z_{l}\) will store \(c^{(l)}\left(\left(u^{\prime}\right)^{p=l}\right)^{\top} u^{p=l}\), where \(c^{(l)}=c_{k}\) if \(p_{k}=l\)
        for \(j \in[n]\) do \(z_{p_{j}} \leftarrow z_{p_{j}}+c_{j} u_{j}^{\prime} u_{j} \quad \triangleright O(n)\) time
        for \(j \in[n]\) do \(C_{j i}^{\prime} \leftarrow-c_{j} /\left(1+z_{p_{j}}\right) \quad \triangleright O(n)\) time
        for \(l \in[r]\) do \(x \leftarrow x+\log \left(1+z_{l}\right) \quad \triangleright O(n)\) time because \(r \leq n\)
        if \(i>0\) then
            \(y \leftarrow \mathbf{0} \in \mathbb{R}^{n \times i} \quad \triangleright O(n i)\) time
            for \(j, k \in[n] \times[i-1]\) do \(y_{p_{j} k} \leftarrow y_{p_{j} k}+u_{j}^{\prime} U_{j k} \quad \triangleright O(n i)\) time
            for \(j, k \in[n] \times[i-1]\) do \(U_{j k}^{\prime} \leftarrow U_{j k}^{\prime}+C_{j i}^{\prime} u_{j}^{\prime} y_{p_{j} k} \quad \triangleright O(n i)\) time
    return \(C^{\prime}, U^{\prime}, x\)
```

matrix, the low-rank matrices do not overlap, and converting an SROS matrix into a Hierarchical matrix would typically be inefficient. Hierarchical matrices admit approximate inversion in $O\left(n a^{2} \log ^{2} n\right)$ time, where $a$ is the maximum rank of the component submatrices (Hackbusch et al., 2004). However, this is not an approximation in a technical sense, as there is no error bound. At many successive steps in the algorithm, a rank $2 a$ matrix is approximated by a rank $a$ matrix (Hackbusch, 1999); to our knowledge there is no analysis of how resulting errors might cascade. After converting an SROS matrix to hierarchical form, this rough inversion would take $O\left(n q^{2} \log ^{2} n\right)$ time.

### 5.5 Binary tree Gaussian process

We now show that our kernel matrix $K_{X X}$ can be written in SROS form, with $P$ containing successively finer partitions. Thus, $K_{X X}$ can be approximately inverted quickly, for use in Equations 5.1 and 5.2. Next, we'll show that we can efficiently optimize the log likelihood of the training data by tuning the weight vector $w$ along with the bit order. The $\log$ likelihood can be calculated in $O(n q \log n)$ time and then the gradient w.r.t. $w$ in $O\left(n q^{2}\right)$ time.

Recall from the proof of Proposition 1: $K_{X X}=\sum_{i=1}^{q} \sum_{s \in \mathbb{B}^{i}} w_{i} X_{[s]} X_{[s]}^{\top}$, where $X_{[s]} \in \mathbb{R}^{n}$ with $\left(X_{[s]}\right)_{j}=\left[\left[X_{j}^{\leq|s|}=s\right]\right]$. So we will set $P_{:, i}, C_{:, i}$, and $U_{:, i}$, so that $L\left(P_{:, i}, C_{:, i}, U_{:, i}\right)=\sum_{s \in \mathbb{B}^{i}} w_{i} X_{[s]} X_{[s]}^{\top}$. Let $P_{:, i}$ partition the set of points $X$ so that points are in the same partition if the first $i$ bits match. Now, requiring the first $i+1$ bits to match is a stricter criterion than requiring the first $i$ bits to match, so the $P_{:, i}$ grow successively finer. For any piece of the partition where the first $i$ bits of the constituent points equals the bitstring $s$, the corresponding sparse rank one component of $K_{X X}$ is $w_{i} X_{[s]} X_{[s]}^{\top}$. So let $U_{:, i}=\mathbf{1}^{n}$, and let $C_{:, i}=w_{i} \mathbf{1}^{n}$.

Proposition 2 (SROS Form Kernel). $K_{X X}=L(P, C, U)$, as defined above.
This follows immediately from the definitions. To compute these partitions $P_{:, i}$, we sort $X$, which is a set of bit strings. And then we can easily compute which points have the same first $i$ bits. This all takes $O(n q \log n)$ time. Now note that $U_{:, i}=U_{:, i^{\prime}}$ for all $i, i^{\prime}$, so $\left(K_{X X}+\lambda I\right)^{-1}$ and $\left|K_{X X}+\lambda I\right|$ can be computed in $O(n q)$ time, rather than $O\left(n q^{2}\right)$.

The training negative log likelihood of a GP is that of the corresponding multivariate Gaussian on the training data. So:

$$
\operatorname{NLL}(w)=\left(y^{\top}\left(K_{X X}(w)+\lambda I\right)^{-1} y+\log \left|K_{X X}(w)+\lambda I\right|+n \log (2 \pi)\right) / 2 .
$$

This can be computed in $O(n q)$ time, since matrix-vector multiplication takes $O(n q)$ time for a matrix in SROS form. So if the bit order is unchanged, an optimization step can be done in $O(n q)$ time, and if the data needs to be resorted, then in $O(n q \log n)$ time. On the largest dataset we tested (House Electric), with $n \approx 1.3$ million and $q=88$, sorting the data and computing $P$ takes about 0.96 seconds on a GPU, and then calculating the negative log likelihood takes about another 1.08 seconds. We show in Appendix A. 8 how to compute $\nabla_{w}$ NLL in $O\left(n q^{2}\right)$ time.

To optimize the bit order and weight vector at the same time, we represent both with a single parameter vector $\theta \in \mathbb{R}_{+}^{q}$, with $\|\theta\|_{\infty}=1$. To get the bit order from $\theta$, we start with a default bit order and permute the bit order according to a permutation that would sort $\theta$ in descending order. To get the weight vector, we sort $\theta$ in descending order, add a 0 at the end, and compute the differences between adjacent elements. When there are ties in the elements of $\theta$, the choice of bit order does not affect the negative log likelihood (or the kernel at all) because the relevant associated weight is 0 . The negative log likelihood is continuous with respect to $\theta$, and when all values of $\theta$ are unique, it is differentiable with respect to $\theta$. Letting $\theta=e^{\phi} /\left\|e^{\phi}\right\|_{\infty}$, we minimize loss w.r.t. $\phi$ using BFGS (Fletcher, 2013).

To calculate the predictive mean at a list of predictive locations $X^{\prime}$, we first multiply $y$ by $\left(K_{X X}+\lambda I\right)^{-1}$, and then we multiply that vector by $K_{X X^{\prime}}$. We obtain both $K_{X X}$ and $K_{X X^{\prime}}$ in SROS form as follows. Let $\tilde{X}=X \circ X^{\prime}$ be the concatenation of the two tuples, now an $(n+m)$-tuple. Writing $K_{\tilde{X} \tilde{X}}=L(\tilde{P}, \tilde{C}, \tilde{U})$, the arrays on the r.h.s. can be computed in $O((n+m) q \log (n+m))$ time. Then, with $P, C$, and $U$ being the first $n$ rows of $\tilde{P}, \tilde{C}, \tilde{U}, K_{X X}=L(P, C, U)$. And letting $P^{\prime \prime}$ and $U^{\prime \prime}$ be the last $m$ rows, $K_{X X^{\prime}}=L\left(P, P^{\prime \prime}, C \odot U, U^{\prime \prime}\right)$. Thus, the predictive mean $\mu_{x}$ from Equation 5.1 can be computed at $m$ locations in $O((n+m) q \log (n+m))$
time.

The predictive covariance matrix, which extends the predictive variance from Equation 5.2, is calculated $\Sigma_{X^{\prime}}=K_{X^{\prime} X^{\prime}}+\lambda I_{m}-K_{X^{\prime} X}\left(K_{X X}+\lambda I_{n}\right)^{-1} K_{X X^{\prime}}=$ $\left(K_{\tilde{X} \tilde{X}}+\lambda I_{m+n}\right) / K_{X X}$, where / denotes the Schur complement. From a property of block matrix inversion, the last $m$ columns of the last $m$ rows of $\left(K_{\tilde{X} \tilde{X}}+\lambda I\right)^{-1}$ equals $\left(\left(K_{\tilde{X} \tilde{X}}+\lambda I_{m+n}\right) / K_{X X}\right)^{-1}$. So we get the predictive precision matrix in $O((n+m) q \log (n+m))$ time by inverting $K_{\tilde{X} \tilde{X}}+\lambda I$ and taking the bottom right $m \times m$ block. Then, we get the predictive covariance matrix by inverting that. This takes $O\left(m q^{2}\right)$ time, since it does not have the property of all the columns of $U$ being equal. If we only want the diagonal elements of an SROS matrix (the independent predictive variances in this case), we can simply sum the rows of $C \odot U \odot U$ in $O(m q)$ time. Thus, in total, computing the independent predictive variances requires $O\left((n+m) q \log (n+m)+m q^{2}\right)$ time. See Algorithm 3 .

### 5.6 Related Work

All existing kernels of which we are aware for linear time GP regression on unstructured data involve inducing points (related to the Nyström approximation (C. Williams \& Seeger, 2000)) or inducing frequencies. For a given set of inducing points $Z$, for some base kernel $k$, the inducing point kernel (in its most basic form) is the following, although subtle variants exist: $k^{Z}\left(x, x^{\prime}\right)=K_{x Z} K_{Z Z}^{-1} K_{Z x^{\prime}}$ (Quinonero-Candela \& Rasmussen, 2005).

Sparse Gaussian Process Regression (SGPR) involves selecting $Z$, and then using $k^{Z}$ (or a variant). Notably, $K_{X X}^{Z}=K_{X Z} K_{Z Z}^{-1} K_{Z X}$ is low rank, providing computational efficiency. The predictive mean and covariance have compact form, with observational noise $\lambda: \mu_{x}(Z)=K_{x Z}\left(\lambda K_{Z Z}+k_{Z X} k_{X Z}\right)^{-1} K_{Z X} y$ and $\sigma_{x x^{\prime}}^{2}(Z)=K_{x Z}\left(K_{Z Z}+\lambda^{-1} k_{Z X} k_{X Z}\right)^{-1} K_{Z x^{\prime}}$.

```
Algorithm 3 GP Regression with a binary tree kernel.
Require: \(X \in \mathbb{B}^{n \times q}, y \in \mathbb{R}^{n}, X^{\prime} \in \mathbb{B}^{m \times q}, w \in \mathbb{R}^{q}, \lambda \in \mathbb{R}^{+}\)
Ensure: \(\mu_{X^{\prime}}\) and \(\sigma_{X^{\prime}}^{2}\) are the predictive means and variances at \(X^{\prime}\), and nll the
    training negative log likelihood.
    \(\tilde{X} \leftarrow X \circ X^{\prime}\)
    \(\tilde{X}^{\uparrow}, \operatorname{perm} \leftarrow \operatorname{Sort}(\tilde{X}) \quad \triangleright\) The rows of \(X\) are sorted lexically
    from leading bit to trailing bit. \(O((n+\)
    \(m) q \log (n+m))\) time.
    for \(j, i \in[n+m] \times[q]\) do \(\tilde{P}_{j i}^{\uparrow} \leftarrow \#\) of unique rows in \(X_{1: j, 1: i}^{\uparrow}\)
        \(\triangleright M_{1: j, 1: i}\) is the first \(j\) rows and \(i\) columns of \(M . O((n+m) q)\) time.
    \(\tilde{P} \leftarrow \operatorname{perm}_{\tilde{P}}{ }^{-1}\left(\tilde{\tilde{P}}^{\uparrow}\right) \quad \triangleright\) This "unsorts" the input. \(O((n+m) q)\) time.
    \(P, P^{\prime} \leftarrow \tilde{P}_{1: n}, \tilde{P}_{n+1: n+m}\)
    \(U, U^{\prime}, \tilde{U} \leftarrow \mathbf{1}^{n \times q}, \mathbf{1}^{m \times q}, \mathbf{1}^{(n+m) \times q}\)
    \(C, \tilde{C} \leftarrow \mathbf{1}^{n} w^{T}, \mathbf{1}^{n+m} w^{T}\)
    \(C_{\lambda}^{-1}, U^{-1}, \operatorname{logdet}_{\lambda} \leftarrow \operatorname{Invert}\left(P, \lambda^{-1} C, U\right) \quad \triangleright\)
    Uses Algorithm 2. Speedup to
    \(O(n q)\) time because columns of \(U\)
    are identical.
    \(C^{-1}, \operatorname{logdet} \leftarrow \lambda^{-1} C_{\lambda}^{-1}, \log ^{2} \operatorname{det}_{\lambda}+n \log (\lambda)\)
    \(z \leftarrow \operatorname{LinTransform}\left(P, P, U^{-1}, C^{-1} \odot U^{-1}, y\right)+\lambda^{-1} y \quad \triangleright\)
    Uses Algorithm 1 to com-
    pute the Woodbury vector.
    \(O(n q)\) time.
\(\mu_{X^{\prime}} \leftarrow \operatorname{LinTransform}\left(P^{\prime}, P, U^{\prime}, C \odot U, z\right) \quad \triangleright O((n+m) q)\) time.
nll \(\leftarrow\left(y^{\top} z+\operatorname{logdet}+n \log (2 \pi)\right) / 2\)
\(\tilde{C}^{\text {prec }}, \tilde{U}^{\text {prec }} \leftarrow \operatorname{Invert}\left(\tilde{P}, \lambda^{-1} \tilde{C}, \lambda^{-1} \tilde{U}\right) \quad \triangleright O((n+m) q)\) time.
\(C^{\text {prec }}, U^{\text {prec }} \leftarrow \tilde{C}_{n+1: n+m}^{\text {prec }}, \tilde{U}_{n+1: n+m}^{\text {prec }}\)
\(C^{\text {cov }}, U^{\text {cov }} \leftarrow \operatorname{Invert}\left(P^{\prime}, C^{\text {prec }}, U^{\text {prec }}\right) \quad \triangleright O\left(m q^{2}\right)\) time; extra factor of \(q\)
                                    because columns of \(U^{\text {prec }}\) are
                                    not identical.
\(\sigma_{X^{\prime}}^{2} \leftarrow \lambda\left(\mathbf{1}^{m}+\operatorname{SumEachRow}\left(C^{\text {cov }} \odot U^{\mathrm{cov}} \odot U^{\text {cov }}\right)\right)\). \(\quad \triangleright O(m q)\) time.
return \(\mu_{X^{\prime}}, \sigma_{X^{\prime}}^{2}\), nll
```

Titsias's (2009a) sparse variational kernel is also low rank and uses inducing points. The sparse variational GP (SVGP) is constructed as the solution to a variational inference problem. It depends on inducing points $Z$, data points $X$, and observed function values $y$. We have focused on Gaussian processes with 0 mean, but the SVGP method uses a nonzero prior mean along with a kernel:
$m^{\operatorname{SVGP}}(x)=\mu_{x}(Z)$ and $k^{\mathrm{SVGP}}\left(x, x^{\prime}\right)=k\left(x, x^{\prime}\right)-K_{x Z} K_{Z Z}^{-1} K_{Z x^{\prime}}+\sigma_{x x^{\prime}}^{2}(Z)$. Given the dependence on $X$ and $y$, this is not a true probability distribution over function space. The variational problem underlying this kernel also provides guidance in how to select the inducing points $Z$. For further discussion of the kernel underlying the SVGP method, see Wild et al. (2021).

An inducing point kernel with $z$ inducing points produces a $z$-dimensional reproducing kernel Hilbert space (RKHS). The dimensionality of the RKHS relates to the expressivity of the kernel. Whereas an inducing point method buys a $z$ dimensional RKHS for the price of $O\left(z^{2} n\right)$ time and $O(z n)$ space, the binary tree kernel produces a $2^{q}$-dimensional RKHS in $O(q n)$ time and space-an exponential improvement. (Observe that we can find $2^{q}$ linearly independent functions of the form $k(\cdot, x)$ —one for each of the $2^{q}$ leaves $x$ might belong to.) Wilson \& Nickisch (2015) develop a method for speeding up inducing point methods significantly, especially in low-dimensional settings.

Lázaro-Gredilla et al. (2010) propose an inducing frequencies kernel: given a set of $m$ inducing vectors $s_{i}, k\left(x, x^{\prime}\right)=1 / m \sum_{i=1}^{m} \cos \left(2 \pi s_{i}^{\top}\left(x-x^{\prime}\right)\right)$. Dutordoir et al. (2020) propose an inducing frequencies kernel for low dimensional data, in which $k\left(x, x^{\prime}\right)$ is a special function of $x^{\top} x^{\prime}$.

On one-dimensional data, filtering/smoothing methods perform Bayesian inference over functions in $O(n)$ time (R. Kalman, 1960; Hartikainen \& Särkkä, 2010). A few non- $O(n)$ methods bear mentioning. We are not the first to consider a kernel over points on the leaves of a tree (Ma \& Blaschko, 2020; Lévesque et al., 2017) or on the leaves of multiple trees (Feng \& Baumgartner, 2020), but their methods take $O\left(n^{3}\right)$ time. On certain kinds of structured data, Toeplitz solvers achieve $O\left(n^{2}\right)$ time complexity (Y. Zhang et al., 2005). Cutajar et al.'s (2016); Wang et al.'s (2019), and others' use of a conjugate gradients solver to replace
inversion/factorization has unclear time complexity between $O\left(n^{2}\right)$ and $O\left(n^{3}\right)$, depending on the kernel matrix spectrum. ${ }^{1}$

### 5.7 Experiments

In Table 5.1, we compare our binary tree kernel and a binary tree ensemble (see Appendix A. 9 for details on the ensemble) against three baseline methods: exact GP regression using a Matérn kernel, sparse Gaussian process regression (SGPR) (Titsias, 2009b), and a stochastic variational Gaussian process (SVGP) (Hensman et al., 2013). We evaluate our method on the same open-access UCI datasets (Dua \& Graff, 2017) as Wang et al. (2019), using their same training, validation, and test partitions, and we compare against the baseline results they report. For the binary tree (BT) kernels, we use $p=\min (8,\lfloor 150 / d\rfloor+1)$, and recall $q=p d$. We set $\lambda=$ $1 / n$. We train the bit order and weights to minimize training NLL. For the binary tree ensemble (BTE), we use 20 kernels. For the Matérn kernel, we use Blackbox Matrix-Matrix multiplication (BBMM) (Gardner et al., 2018), which uses the conjugate gradients method to calculate matrix-vector products with $\left(K_{X X}+\lambda I\right)^{-1}$. SGPR uses 512 data points and SVGP uses 1,024 inducing points. We report the mean and two standard errors across 3 replications with different dataset splits. For further experimental details, see Appendix A.9. BTE achieves the best NLL on $6 / 12$ datasets, and best RMSE on $5 / 12$ datasets (including some ties). Out of the 4 largest datasets, BT/BTE is fastest on 3 . The run times are plotted in Figure 5.5. The code is available at https://github.com/mkc1000/btgp and

[^0]| Dataset | $n$ | $d$ | BTE | BT | MATÉRN (BBMM) | SGPR | SVGP |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| PoleTele | 9,600 | 26 | -0.625 $\pm 0.035$ | $-0.490 \pm 0.040$ | $-0.180 \pm 0.036$ | $-0.094 \pm 0.008$ | $-0.001 \pm 0.008$ |
| Elevators | 10,623 | 18 | $0.649 \pm 0.032$ | $0.646 \pm 0.023$ | $0.619 \pm 0.054$ | $0.580 \pm 0.060$ | $0.519 \pm 0.022$ |
| Bike | 11,122 | 17 | $-0.708 \pm 0.433$ | $-0.806 \pm 0.273$ | $0.119 \pm 0.044$ | $0.291 \pm 0.032$ | $0.272 \pm 0.018$ |
| Kin40K | 25,600 | 8 | $0.869 \pm 0.004$ | $0.881 \pm 0.008$ | -0.258士0.084 | $0.087 \pm 0.067$ | $0.236 \pm 0.077$ |
| Protein | 29,267 | 9 | $\mathbf{0 . 7 8 1} \pm 0.023$ | $0.845 \pm 0.026$ | $1.018 \pm 0.056$ | $0.970 \pm 0.010$ | $1.035 \pm 0.006$ |
| KeggDir | 31,248 | 20 | $-1.031 \pm 0.020$ | $-1.029 \pm 0.021$ | $-0.199 \pm 0.381$ | $\mathbf{- 1 . 1 2 3} \pm 0.016$ | $-0.940 \pm 0.020$ |
| CTslice | 34,240 | 385 | $\mathbf{- 2 . 5 2 7} \pm 0.147$ | $-1.092 \pm 0.147$ | $-0.894 \pm 0.188$ | $-0.073 \pm 0.097$ | $1.422 \pm 0.005$ |
| KEGGU | 40,708 | 27 | $-0.667 \pm 0.007$ | $-0.667 \pm 0.007$ | $-0.419 \pm 0.027$ | $-0.984 \pm 0.012$ | $-0.666 \pm 0.007$ |
| 3DROAD | 278,319 | 3 | -0.251 $\pm 0.009$ | -0.252 $\pm 0.006$ | $0.909 \pm 0.001$ | $0.943 \pm 0.002$ | $0.697 \pm 0.002$ |
| Song | 329,820 | 90 | $1.330 \pm 0.003$ | $1.331 \pm 0.003$ | $1.206 \pm 0.024$ | $1.213 \pm 0.003$ | $1.417 \pm 0.000$ |
| Buzz | 373,280 | 77 | $1.198 \pm 0.003$ | $1.198 \pm 0.003$ | $0.267 \pm 0.028$ | $\mathbf{0 . 1 0 6} \pm 0.008$ | $0.224 \pm 0.050$ |
| HouseElec | 1,311,539 | 11 | $\mathbf{- 2 . 5 6 9} \pm 0.006$ | $-2.492 \pm 0.012$ | $-0.152 \pm 0.001$ | - | $-1.010 \pm 0.039$ |
| PoleTele | 9,600 | 26 | $\mathbf{0 . 1 5 4} \pm 0.006$ | $0.161 \pm 0.004$ | $\mathbf{0 . 1 5 1} \pm 0.012$ | $0.217 \pm 0.002$ | $0.215 \pm 0.002$ |
| Elevators | 10,623 | 18 | $0.478 \pm 0.021$ | $0.476 \pm 0.018$ | $\mathbf{0 . 3 9 4} \pm 0.006$ | $0.437 \pm 0.018$ | $0.399 \pm 0.009$ |
| Bike | 11,122 | 17 | $\mathbf{0 . 1 1 8} \pm 0.057$ | $0.103 \pm 0.029$ | $0.220 \pm 0.002$ | $0.362 \pm 0.004$ | $0.303 \pm 0.004$ |
| Kin40K | 25,600 | 8 | $0.580 \pm 0.003$ | $0.587 \pm 0.006$ | $0.099 \pm 0.001$ | $0.273 \pm 0.025$ | $0.268 \pm 0.022$ |
| Protein | 29,267 | 9 | $0.608 \pm 0.008$ | $0.623 \pm 0.011$ | $0.536 \pm 0.012$ | $0.656 \pm 0.010$ | $0.668 \pm 0.005$ |
| KeggDir | 31,248 | 20 | $0.086 \pm 0.003$ | $0.086 \pm 0.003$ | $0.086 \pm 0.005$ | $0.104 \pm 0.003$ | $0.096 \pm 0.001$ |
| CTslice | 34,240 | 385 | $\mathbf{0 . 1 1 6} \pm 0.009$ | $0.132 \pm 0.009$ | $0.262 \pm 0.448$ | $0.218 \pm 0.011$ | $1.003 \pm 0.005$ |
| KEGGU | 40,708 | 27 | $0.120 \pm 0.001$ | $0.121 \pm 0.001$ | $0.118 \pm 0.000$ | $0.130 \pm 0.001$ | $0.124 \pm 0.002$ |
| 3DROAD | 278,319 | 3 | $0.187 \pm 0.002$ | $0.186 \pm 0.001$ | $0.101 \pm 0.007$ | $0.661 \pm 0.010$ | $0.481 \pm 0.002$ |
| Song | 329,820 | 90 | $0.914 \pm 0.003$ | $0.916 \pm 0.003$ | $0.807 \pm 0.024$ | $0.803 \pm 0.002$ | $0.998 \pm 0.000$ |
| Buzz | 373,280 | 77 | $0.801 \pm 0.002$ | $0.801 \pm 0.002$ | $0.288 \pm 0.018$ | $0.300 \pm 0.004$ | $\mathbf{0 . 3 0 4} \pm 0.012$ |
| HouseElec | 1,311,539 | 11 | $0.029 \pm 0.001$ | $0.029 \pm 0.001$ | $0.055 \pm 0.000$ |  | $0.084 \pm 0.005$ |
| PoleTele | 9,600 | 26 | $5.16 \pm 0.58$ |  | $0.69 \pm 0.018$ | $1.16 \pm 0.34$ | $1.15 \pm 0.068$ |
| Elevators | 10,623 | 18 | $2.6 \pm 0.19$ |  | $0.68 \pm 0.012$ | $1.16 \pm 0.38$ | $1.27 \pm 0.092$ |
| Bike | 11,122 | 17 | $2.68 \pm 0.15$ |  | $0.69 \pm 0.015$ | $1.17 \pm 0.38$ | $1.28 \pm 0.093$ |
| Kin40k | 25,600 | 8 | $1.44 \pm 0.028$ |  | $0.71 \pm 0.045$ | $1.62 \pm 0.96$ | $3.26 \pm 0.23$ |
| Protein | 29,267 | 9 | $2.92 \pm 0.2$ |  | $\mathbf{0 . 8} \pm 0.17$ | $2.27 \pm 0.9$ | $3.31 \pm 0.27$ |
| KeggDir | 31,248 | 20 | $7.14 \pm 0.39$ |  | $\mathbf{0 . 8 5} \pm 0.1$ | $2.2 \pm 1.09$ | $3.8 \pm 0.38$ |
| CTslice | 34,240 | 385 | $52.01 \pm 0.92$ |  | $3.32 \pm 5.0$ | $\mathbf{2 . 1 6} \pm 0.99$ | $3.87 \pm 0.34$ |
| KEGGU | 40,708 | 27 | $7.46 \pm 0.54$ |  | $6.32^{*} \pm 0.41^{*}$ | $\mathbf{2 . 2 2} \pm 1.05$ | $4.78 \pm 0.4$ |
| 3DROAD | 278,319 | 3 | $1.93 \pm 0.12$ |  | 126.37* $\pm 20.92^{*}$ | $12.01 \pm 5.51$ | $34.09 \pm 3.19$ |
| Song | 329,820 | 90 | $31.87 \pm 3.79$ |  | $33.79^{*} \pm 10.45 *$ | $7.89 \pm 3.12$ | $39.55 \pm 3.08$ |
| Buzz | 373,280 | 77 | $\mathbf{2 0 . 1 8} \pm 6.66$ |  | $571.15 * \pm 66.34^{*}$ | $\mathbf{2 9 . 2 5} \pm 18.33$ | $46.35 \pm 2.93$ |
| HouseElec | 1,311,539 | 11 | $118.41 \pm 3.93$ |  | $575.64 * \pm 6.94 *$ | - | $367.71 \pm 4.7$ |

Table 5.1: NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM2-16GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU. All columns except BT and BTE come from Wang et al. (2019).
https://tinyurl.com/btgp-colab.

BT performs noticeably worse than BTE for test NLL on CTSlice due to overfitting. There are enough degrees of freedom when optimizing the bit order ( $d=$ 385) that BT kernel can over-fit to the training data. The ensemble over multiple bit orders is much more robust.


Figure 5.5: Run times given dataset size. For BT, the trendline is calculated controlling for $\log (q)$ with affine regression, and then setting $q=150$. The slope w.r.t. $\log (q)$ is $2.82 \pm 1.06$. Theoretically, all slopes are too low except for that of SVGP, presumably because of overhead in the small-data regime.

### 5.8 Discussion

We have proven that the binary tree kernel GP is scalable. Our empirical results suggest that it often outpredicts not just other scalable methods, but even the popular Matérn GP. If the results in this paper replicate in other domains, it could obviate wide usage of classic GP kernels like the Matérn kernel, as well as inducing point kernels. Sometimes, our kernel fails to capture patterns in the data; some functions' values simply do not covary this way. But other kernels we tested seemed to fail like that even more.

Our contributions to linear algebra and kernel design may significantly increase the size of data sets on which GPs can do state-of-the-art modelling.

### 5.9 End notes

Some closing comments: the only way in which the quality of the uncertainty estimates is tested is through the negative log-likelihood. If the model gives too wide a confidence interval, it will assign less marginal probability to the truth than it could, and if it gives too narrow a confidence interval such that the truth is way outside, then it will be seriously penalized for assigning (perhaps vanishingly) small marginal probability to the truth. So a good NLL score does imply good uncertainty quantification, but only very roughly. Maybe some estimates are extremely precise and accurate, contributing to a low average NLL, while others are fairly sloppy. So future work would be helpful to get a clearer picture of the quality of the uncertainty estimates. For example, a calibration analysis would dissect the origin of the strong NLL performance, and check just how reliable or how hit-and-miss the uncertainty quantification is.

Some other GP methods that may be of interest to the reader are ones that a) are fast for low dimensional data, or b) use a kernel that is separable over its dimensions, which enables similar speed. Särkkä (2013) discusses state space GPs, which track sufficient statistics that eliminate the need to handle the entire covariance matrix. In the simplest case, one dimensional Gaussian processes are Markovian, so they can be reduced to a Kalman filtering problem (Hartikainen \& Särkkä, 2010). For state space GPs in few dimensions, see e.g. Särkkä \& Hartikainen (2012). Unfortunately these methods don't scale to higher dimensions. Another approach for low-dimensional data is Hensman et al.'s (2017) Variational Fourier Features. This method takes inspiration from Rahimi \& Recht's (2007) Random Fourier Features, but then optimizes those features according to a variational objective. This amounts to selecting inducing frequencies instead of inducing points. Unfortunately, for general kernels, an approximation of the base kernel
(to within some tolerance) requires a number of frequencies that is exponential in the dimension of the data. Another method that is natural for low-dimensional data is Samo \& Roberts's (2016) String GP, which involves partitioning the domain into intervals whose interdependence is screened off by the interfaces. All of these methods can be extended to a high-dimensional domain if the kernel can be written as a sum of one-dimensional kernels.

Another kind of kernel that may interest the reader is a wavelet kernel (L. Zhang et al., 2004), since the binary tree kernel looks like a sum of "pulses" which resemble wavelets. Unfortunately, while a wavelet GP can approximate arbitrary functions, unlike a binary tree kernel, it does not offer any speedup in run time compared to normal kernels.

Finally, I'll spend a bit more time on the parameters and hyperparameters of this method. As discussed in Section 5.5 (see the definitions of $\theta$ and $\phi$ ), there is a parameter for each bit, giving a total of $q$ parameters. Besides what value of $q$ to use, the only hyperparameters are those of the optimization method, like the learning rate, the number of initializations, etc. For $q$, as we show in Appendix A.10, increasing it improves performance, so best to increase it if one has the time. So there is no real difficulty in setting adequate hyperparameters, and of course the parameters are set through optimization. In our experiments, we use lower precision as dimensions increase to target a $q$ of around 150. Perhaps this is a large number of parameters, but we do not seem to observe much over fitting; perhaps this is because parameters corresponding to very "deep" bits don't have much impact on the kernel.

## 6| Literature Review

The bodies of work relevant to each chapter are fairly disjoint, and to avoid rehearsing the literature reviews of each of the papers, I'll be stepping back quite a bit to give some further context to the work. The related work of primary relevance to this thesis can be found in the related work sections of the individual chapters. I'll first review work related to the expected behavior of advanced agents. Then I'll discuss Hutter's (2005) AIXI, to which the pessimistic reinforcement learner is closely related, and to which the pessimistic imitation learner is somewhat related. Finally, to give some context to my work on Gaussian processes, I'll discuss Bayesian methods in general.

### 6.1 Expected Behavior of Advanced Agents

Bostrom (2014) and Russell (2019) have written at book-length about why advanced artificial agents might pose a serious danger to us. On the specific problem of an agent intervening in the provision of goal-information, this has been discussed in the literature using various terms-wireheading, reward hacking, reward tampering, and delusion-boxing. The term wireheading is inspired by an experiment in which rats repeatedly pressed a lever that directly stimulated a "happiness" neuron in their brain (Olds, 1958). It has been widely discussed (Bostrom, 2014; Amodei et al., 2016; Taylor et al., 2016; Russell, 2019; Kumar et al., 2020; Everitt et al., 2021; Ring \& Orseau, 2011).

There is also existing work on ensuring that once an agent has received a percept, the way it processes that percept is not altered (Everitt et al., 2016, 2021). This has also been called wireheading, which is why we avoid the term. That problem (as those papers show) is easily soluble. The problem that concerns me is an artificial
agent interrupting the protocol by which we intended to provide percepts, including goal-informative percepts like rewards; (Everitt et al. (2021) only consider this in the setting of a known environment).

In my opinion, there is no existing work that address the problem of reward-provision-intervention head on, except arguably the contents of Chapter 3, as explained in Section A.4.2. But there is other work that sidesteps this problem. Imitation learning can produce intelligent behavior (if the demonstrator is intelligent) without any algorithmic conception of a goal. We review the imitation learning literature more extensively below. Taylor (2016) conceives of a hybrid between imitation and optimization called quantilization. Quantilizers imitate a demonstrator, conditioned on the demonstrator performing better than they do the (vast) majority of the time. "Better" is with respect to some goal like rewardmaximization. Depending on how vast "vast" is, quantilization can look more like imitation or more like optimization. There is also work on how we might make an advanced artificial agent that is unable to intervene in the provision of reward (M. K. Cohen et al., 2020). And there is plenty of work on myopic agents, which might not have time to intervene in the provision of reward; in the case of extreme myopia, these are known as (contextual) bandits (Lattimore \& Szepesvári, 2020). A deeper review of the bandit literature would be very tangential, so I'll direct the interested reader to Lattimore \& Szepesvári (2020).

### 6.2 Robust Control and Safe RL

Reinforcement learners that are not particularly advanced can also exhibit dangerous misbehavior by accident. Robust control and (so-called) Safe RL both attempt to mitigate such errors.

Robust control is one of the classic computer science problems that predates com-
puters. Much work in the field describes the state as a vector in a vector space, and various assumptions are made about how the state evolves. A celebrated result by R. E. Kalman (1960) derives an optimal policy assuming that the state evolves according to a linear transform and that the cost function is quadratic. And the discrete Kalman filter (R. Kalman, 1960), which allows for noise but restricts to discrete time, was used to land on the moon for the first time. Outer space is rife with unsafe states, especially when you have limited fuel. $H_{\infty}$ control is perhaps the most robust version of control in the standard paradigm, and has been explored by Tannenbaum (1980); Zames (1981); Barbu \& Sritharan (1998); Simon (2006).

For a historical review dating back to 1927, including myriad extensions with different assumptions, see Dorato (1987). For a more modern treatment, any number of textbooks cover the topic extensively, like Dullerud \& Paganini (2013).

As the assumptions on the environment relax more and more, especially allowing for the environment to be unknown, robust control starts to be called safe RL. This is often accompanied by a shift in notational heritage. For an extensive review, see García \& Fernández (2015). There are two key branches of Safe RL: avoiding the risks from misdirected exploitation and the risks from exploration. Risk-aware optimization criteria, which address risks from misdirected exploitation, are covered in the literature review from Chapter 3, as these are most closely related to pessimism. Another class of approach is to constrain the policy in some way. For example, Moldovan \& Abbeel (2012) require the policy only approach states from which it is possible to return to the start. For certain practical problems with known constraints, policy contraints can be hand-designed. For example, Abe et al.'s (2010) work, entitled "Optimizing debt collections using constrained reinforcement learning", examines how certain real-world constraints (like legal ones) can be incorporated into an RL problem. If one has access to a complete list of dangerous states, one can use method like Polymenakos et al.'s (2019).

There have been many attempts to design safe exploration. But the difficulty is fundamental. Exploration necessarily involves visiting states before you know if they're good or bad. What if they're bad?! One work-around is to ask for help. That's our approach to exploration in Chapter 3. It's also investigated by Clouse (1997), Hans et al. (2008), García \& Fernández (2012), and García et al. (2013). Relatedly, a human monitor can provide unsolicited help at times (Clouse \& Utgoff, 1992; Maclin \& Shavlik, 1996; Thomaz \& Breazeal, 2006).

A key property of the pessimistic agent in Chapter 3 is that it sometimes abstains from taking an action and asks for help. This abstention recalls KWIK learning ("knows what it knows") L. Li et al. (2008). KWIK learning has been incorporated into RL agents in many ways (T. J. Walsh et al., 2009; T. Walsh et al., 2010; Szita \& Szepesvári, 2011; Lang et al., 2012); usually when they know that they don't know something, they will explore in some way, but they could just as easily be made to ask for help in those situations instead.

Some methods attempt to do safe exploration without help. One is from Gehring \& Precup (2013); their agent avoids exploring states that it predicts will produce a hard-to-control situation. These attempted generalizations don't come with any guarantees, of course, but they're better than nothing. And this method doesn't preclude all exploration. Turchetta et al. (2016) design an agent that takes actions that probably lead to states that are "similar" to observed states. Pan et al. (2017) test the intuitive idea of exploring within a simulation before acting in reality.

### 6.3 AIXI

Instrumental rationality is the project of picking actions in the service of goals. An agent takes actions and receives observations. A world-model is a probability distribution over the next observation given the interaction history so far. There
is much interest in world-models that satisfy a Markov assumption: the probability distribution over the next observation only depends on the latest observation (which is then called the 'state') and the latest action, but this assumption is not core to instrumental rationality.

The most interesting agents act to accomplish goals, which are a function of their observations. The simplest way to make an agent with a goal is to separate the observation into an observation and 'reward', and then design the agent to optimize reward. Perfect planning with respect to a world-model is expectimax planning, and perfect inference in Bayesian inference. These pieces compose Hutter's (2005) formalism for general intelligence, called AIXI.

Using a countable class of world-models $\mathcal{M}$, a prior weight $w(\nu)>0$ for $\nu \in \mathcal{M}$, and the corresponding Bayes-mixture world-model $\xi=\sum_{\nu \in \mathcal{M}} w(\nu) \nu$,

$$
\begin{equation*}
\pi^{\mathrm{AIXI}}=\underset{\pi \in \Pi}{\operatorname{argmax}} \mathbb{E}_{\xi}^{\pi} \sum_{t=1}^{m} r_{t} \tag{6.1}
\end{equation*}
$$

where $\Pi$ is the set of policies that depend on the entire interaction history, $\mathbb{E}_{\xi}^{\pi}$ denotes that actions are sampled from $\pi$ and observations and rewards from $\xi$, and $m$ is a horizon length.

When $\mathcal{M}$ contains the truth, and the prior on it is not-minuscule, this agent would likely be generally intelligent. Thus, AIXI is typically constructed with a particularly powerful model class and prior: the class of semicomputable world-models and the Solomonoff prior (Solomonoff, 1964) $w(\nu)=2^{-K(\nu)}$, where $K(\nu)$ is the Kolmogorov complexity, or the length of the shortest program that computes the index of the world-model in an ennumeration of the semicomputable worldmodels. See M. Li \& Vitányi (2008) for the intricacies of Kolmogorov Complexity. If the objective probabilities of events in our universe are semicomputable
(computably approximable with a monotonically increasing sequence of approximations), then this model class contains the truth.

Given whatever prior beliefs about the world one has before seeing anything, if AIXI is constructed to have those prior beliefs, AIXI defines optimal uninformed reward acquisition.

The pessimistic reinforcement learner of Chapter 3 is very similar to AIXI, except instead of taking a Bayes mixture over all world models, the pessimistic agent looks for the worst-case model among a set of models of high posterior weight.

The pessimistic imitation learner is obviously not trying to maximize any reward like AIXI is. But a key similarity is that when learning, the environment never resets. As mentioned in the imitation learning paper, all previous work on imitation learning, to my knowledge, regards a setting where the environment resets during learning. This is also the norm in the reinforcement learning literature. But the AIXI formalism recognizes that full generality requires removing that handicap. The other key inheritance from AIXI is a discrete model class over which the algorithm does full Bayesian updating.

AIXI "explores" with value-of-information calculus. I add scare quotes, because in a sense, it is just enlightened exploitation. Because of this, in some circumstances, AIXI never explores certain slim possibilities. Suppose that, according to AIXI's posterior, taking action $a_{0}$ always has expected value $1 / 2$, whereas it has $2 / 3$ credence in a world-model that implies that taking action $a_{1}$ locks in a reward of 0 forever, and $1 / 3$ credence in a world-model that implies that it locks in reward 1 forever. It is not worth testing action $a_{1}$ to learn. Arguably, this is a feature not a bug, if those slim possibilities really aren't worth exploring. However, it does mean that AIXI has no formal performance guarantees besides the somewhat unsatisfying optimality-by-definition.

Some "solutions" to this "problem" have been presented (Lattimore \& Hutter, 2014a; Leike, 2016; Leike, Lattimore, et al., 2016; M. K. Cohen et al., 2019). They ensure optimality in the limit. However, M. K. Cohen et al. (2021) show that the exploration required for such an optimality guarantee (in general computable environments) ensures that the reinforcement learner becomes destroyed or incapacitated. (Note that you never act suboptimally once you are dead). This explains the choice to have the pessimistic reinforcement learner outsource its exploration to a human mentor. And it means that the theoretical performance guarantees of M. K. Cohen \& Hutter (2020) and M. K. Cohen, Hutter, \& Nanda (2022) are actually about as strong as one can hope for.

### 6.4 Bayesian Methods

The binary tree kernel of Chapter 5 gives us method for doing Gaussian processes regression. Gaussian processes (C. K. Williams \& Rasmussen, 2006) are an example of a Bayesian machine learning method. In the paper's literature review section, I review other methods for making GPs more tractable. Here I will discuss Bayesian methods more broadly.

Bayes' Rule (Bayes, 1763) follows in a few steps from the definition of conditional probability; nowadays, it takes more work to have a mathematical object bear your name. It states

$$
\begin{equation*}
\mathrm{P}(\text { hypothesis } \mid \text { observations })=\frac{\mathrm{P}(\text { hypothesis }) \mathrm{P}(\text { observations } \mid \text { hypothesis })}{\mathrm{P}(\text { observations })} \tag{6.2}
\end{equation*}
$$

One of the most basic inference algorithms is linear regression, and one of the most basic Bayesian inference algorithms is Bayesian linear regression (Murphy, 2012). In Bayesian linear regression, one starts with a prior probability distribu-
tion over each of the coefficients in a linear model, and then this is updated into a posterior after seeing the data. If the prior distributions are all Gaussian, this is computationally easy. Bayesian linear regression can be made more powerful by transforming the data first, often into a higher-dimensional space. For example, Lázaro-Gredilla \& Figueiras-Vidal (2010) and Ober \& Rasmussen (2019) transform data according to the activations of the neurons in the last hidden layer of a neural network, and then perform Bayesian linear regression.

In Gaussian process regression, each hypothesis is a function, and the observations are (noisy) observations of the function's true value at various locations. Amazingly, given the enormous support of the Gaussian process prior, the posterior over functions can be finitely described, and a posterior over the function's value at any given location can be computed. In fact, it is not trivial to show that Gaussian processes exist at all (Doob, 1953).

Gaussian process regression and Bayesian linear regression (with a Gaussian prior) are closely related-any GP with a finite dimensional kernel can be recast as Bayesian linear regression (C. K. Williams \& Rasmussen, 2006).

Another successful branch of Bayesian machine learning is probabilistic graphical modelling (Koller \& Friedman, 2009). In probabilistic graphical models, a joint distribution over many variables can be concisely defined in terms of the joint marginal distributions over "neighboring" variables' values. A graph is drawn to determine which variables are neighbors. The book just cited gives an extensive account of the techniques and uses of probabilistic graphical models.

Unfortunately, for most model classes, for most prior distributions that we are interested in, the posterior distribution after seeing the data has no simple representation. So a large part of the field of Bayesian machine learning is about approximating Bayesian inference. The two most common techniques (in my
impression) are variational approximations and Markov chain Monte Carlo sampling.

Variational inference is a technique in which a posterior distribution is (locally) approximated by a distribution that is more tractable to represent and evaluate. This is often done by attempting to minimize the KL divergence from the approximate model to the truth. This is easier said than done. Another formulation of this is to maximize the evidence lower bound, or ELBO (Mcauliffe \& Blei, 2007). For a true distribution $p_{\theta}$ over observables $x$ and latent variables $z$, we would often like to marginalize over the latent variables. And the evidence lower bound is that $p_{\theta}(x)=\int p_{\theta}(x, z) d z \geq \mathbb{E}_{z \sim q_{\phi}} \ln \frac{p_{\theta}(x, z)}{q_{\phi}(z)} . q_{\phi}$ can be any distribution, so we pick one that is easy to evaluate and sample from. Because the expectation is over an easy-to-sample-from distribution, it can be approximated with samples.

When picking tractable distributions to appproximate the true distribution, it is very helpful to pick a "conjugate prior". Such a prior distribution produces a posterior distribution with analytic form in the same family of distributions after a certain kind of evidence comes in. See for example, Hensman et al.'s (2012) use of the conjugate exponential family for variational inference.

For a further review of variational inference, see Blei et al. (2017) and C. Zhang et al. (2018).

While a true posterior distribution is intractable in many contexts, an unnormalized version is often accessible. Markov chain Monte Carlo sampling is a method for sampling from an unnormalized posterior distribution Gilks et al. (1995). The insight behind the method comes from the stationary distribution of states in a Markov chain. Because the relative probability of states is known from the unnormalized posterior, an appropriate transition kernel can be constructed for which the stationary distribution matches the true posterior (Metropolis et al., 1953;

Hastings, 1970). The transition kernel works as follows: sample a new state according to a symmetric distribution (sampling $b$ from $a$ has the same probability as sampling $a$ from $b$ ), but then with some probability, stay put instead. If the unnormalized posterior at the new location is less than that of the current location, the probability of moving is equal to that ratio. This works, but relatively slowly. Gilks et al. (1995) covers the extensive literature on the problem of how to design a different transition kernel to make this kind of method converge more quickly.

Finally, I'll discuss a more recent proposal for approximating Bayesian inference: just learn to do it heuristically from practice in lots of settings. Ortega et al. (2019) discuss how to do this for a sequential prediction task. The Bayes-optimal way to do sequential prediction is to keep running posterior credences over hypotheses as each token of the sequence comes in. Notably, the posterior credences at any point in time form a "sufficient statistic"-they have all the information you need for predicting going forward, so you can forget about what the actual observations were. So the problem of updating sufficient statistics given observed data, and then using those statistics to make predictions is best solved by exact (amortized) Bayesian inference (Ritchie et al., 2016), but this is often intractable; maybe a learned function could quickly approximate this computation. Ortega et al. (2019), Genewein et al. (2023, forthcoming), and Kirsch et al. (2022) train a function through meta-learning to do this on sequences generated in qualitatively diverse ways. To succeed, this function must do something like Bayesian inference. The proposal that meta-learners approximate Bayesian learning is recent, but meta-learning itself has been studied for a while (Bengio et al., 1990; Schmidhuber et al., 1996; Thrun \& Pratt, 1998).

For many other Bayesian methods and much more detail on the ones discussed above, see Murphy (2012) and MacKay (2003).

### 6.5 Ensembling

Of the many methods in machine learning to quantify uncertainty (Psaros et al., 2023), the most widespread strategy outside of pseudo-Bayesian methods appears to be ensembling (Dietterich, 2000). In fact, ensembling appears in every technical chapter of this thesis, as I'll explain in a moment. An ensemble is a set of multiple different models, and one can quantify uncertainty at a test location by examining the extent to which their predictions agree. A key question for ensembling methods is how the models in the ensemble are encouraged to differ from each other.

In Chapter 5, the ensemble is named as such. The binary tree ensemble is made of many different binary tree GPs, and after the predictive distributions are added together, the resulting accuracy usually increases. The only source of difference between the models is different initialization of the optimization process, and the highly non-convex optimization surface. In Chapters 3 and 4, the "top set" of models is an ensemble. And the output of both algorithms depend critically on how much the models in the ensemble agree. The way that variety is encouraged in these ensembles is that we start with an extensive variety of models a priori, and all models that perform well enough are allowed in.

A classic method for promoting variety in an ensemble is bootstrapping (Efron, 1979). In this method, each model is trained on a new dataset that is made out of data points that are sampled (with replacement) from the original dataset. For some methods, ensemble variety can be forced by making different algorithmic choices. For example, in Breiman's (2001) random forest-an ensemble of decision trees-each tree uses a different order of the dimensions along which the data is successively split.

When training a neural network, random initialization of the parameters and random shuffling of the data seems to suffice for ensemble variety (Lakshminarayanan et al., 2017). Wenzel et al. (2020) show that ensembling over different hyperparameters as well can improve performance. These and other methods for quantifying uncertainty in neural networks have been evaluated by Nado et al. (2021).

Naïvely, an ensemble with 100 models has a 100x cost. So there has been some work to create ensembles of neural networks that can be trained and evaluated more efficiently. Wen et al. (2020) create an ensemble of neural networks where each weight matrix in a network is Hadamard-multiplied by different rank one matrices to get alternative networks in the ensemble. With comparable performance, the resource requirements are significantly lower, especially as the number of models grows. Osband et al. (2021) develop a class of models called Epistemic Neural Networks and a particular architecture called an epinet. The epinet effectively creates a single network than can act differently depending on a different random input (separate from the ordinary input), so one network represents many models in an ensemble. The models also have different weight depending on the marginal probability of the random input. This method makes training way more efficient than a large ensemble, but it also outperforms them. More recently, they were evaluated in their ability to accommodate distributional shift, and they outperformed standard ensembles (Lu et al., 2022).

## 7| Conclusion

I have argued that advanced agents planning over the long-term in unknown environments present an existential threat to humanity. I have shown that in theory, a sufficiently pessimistic agent would not present such a threat, nor would a sufficiently "pessimistic" imitation learner (although imitation learners are not so threatening to begin with). Then, I developed a new log-linear-time method for Gaussian process regression in order to push forward the state-of-the-art in artificial uncertainty, a prerequisite for artificial pessimism.

I hope this work opens a rich space of ideas for future work to explore. The algorithms from Chapters 3 and 4 are not yet ready for practical application, but the ideas are. The possibilities for how these algorithms might be tractably approximated are endless, and then they have the potential to be be extremely useful. For example, the set of top models $\left(\mathcal{M}_{t}^{\beta}\right.$ in Chapter $3 \Pi_{h_{<t}}^{\alpha}$ in Chapter 4) could be approximated by an ensemble. And the existing literature on how to promote diversity among an ensemble could be brought to bear on this problem. Or the sets could be approximated by repeated sampling from a Bayesian posterior; the reader is invited to bring their favorite (approximate) Bayesian method to the problem. Note that only various minima over the ensemble affect the final decisions in Chapters 3 and 4, and the rest of the ensemble is ignored. So another approach would be to, for each action, search for demonstrator-models that assign low probability that action; or for each candidate policy, search for worldmodels that assign low value to that policy. These searches would of course need to be balanced by the search for accurate models. In the neural network paradigm, where model search is done with gradient descent, recent work by Rigter et al. (2022) takes something like this approach. I cannot say their work was inspired by mine, but if it had been, it might have looked quite similar; there are many
possible choices that could be made along the way.
How to test these ideas empirically? For pessimistic reinforcement learning, one could try to create an agent that learns to land a lunar lander without crashing once during training, with the help of a fuel-inefficient mentor who is nonetheless able to avoid crashing. Could the pessimistic agent retain the no-crashing record, wean itself from mentorship, and surpass its mentor in fuel-efficiency? For pessimistic imitation learning, if the true demonstrator's policy is known (to the experimenters), one could evaluate the imitator's ability to assign low probability to very low probability actions. For example, a large language model obviously isn't a person, but if one imagined that the large language model was what you cared about imitating very carefully, that could be used as a demonstrator. (And presumably, if the imitator did well at that task, it could be similarly careful when imitating a real person).

It's interesting to see that Rigter et al. (2022) test their agent on offline RL benchmarks. So a) it would be interesting to see whether an agent like theirs could perform well at the lunar lander task above. And b) maybe offline RL benchmarks are a good testing suite to encourage cautious agent design.

Chapter 4 could also enable an invigoration of a classic use of an imitation learner: conditioning an imitative policy on a future desired outcome. Such conditioning can put extraordinary strain on the accuracy of the underlying policy. A perfect Bayesian imitation learner will occasionally assign probability to actions that the demonstrator would never take, because it entertains (obscure) demonstrator models that say the demonstrator would take that action. This can be thought of as epistemic humility. But if this Bayesian imitative policy is conditioned on an unlikely future outcome, a large fraction of the measure on this outcome may come from actions the demonstrator would never take, from the contribution of
erroneous demonstrator models, entertained only out of humility. So any intuition we might have about a conditioned imitative policy still resembling the demonstrator fails. With the method in Chapter 4, however, the policy could be safely conditioned on future events. Empirical evaluation of this approach may need to wait until there are better tractable approximations of this method.

Future work on the Binary Tree GP could involve much more extensive testing and application. It would be worth examining calibration curves in various contexts, the benefits of ensembling, and the possibility of overfitting. It would be interesting to examine the shape of the loss surface in detail. In terms of applications, I've wondered if it could be used to beat the market. The prices of many other stocks could make stock price prediction an arbitrarily high-dimensional problem. A world-model based on a BTGP could be used to help an artificial agent handle uncertainty, as was my original motivation for this method.

This thesis suggests that in theory, in the presence of mentorship, pessimism is a viable strategy for acting safely and competently. I hope to see other work exploring how we might implement practical versions of pessimistic agency. It may require foundational progress in learning models that have high-quality uncertainty estimates.

## Bibliography

Abbeel, P., \& Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In Proceedings of the twenty-first international conference on machine learning (p. 1).

Abe, N., Melville, P., Pendus, C., Reddy, C. K., Jensen, D. L., Thomas, V. P., ... Kowalczyk, M. (2010). Optimizing debt collections using constrained reinforcement learning. In Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 75-84).

Adams, S., Cody, T., \& Beling, P. A. (2022). A survey of inverse reinforcement learning. Artificial Intelligence Review, 1-40.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., \& Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

Aslund, H., Mhamdi, E. M. E., Guerraoui, R., \& Maurer, A. (2018). Virtuously safe reinforcement learning. arXiv preprint arXiv:1805.11447.

Barbu, V., \& Sritharan, S. S. (1998). H-infinity control of fluid dynamics. Proc $R$ Soc Lond Ser A, 545, 3009-3033.

Bayes, T. (1763). LII. An essay towards solving a problem in the doctrine of chances. by the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. Philosophical transactions of the Royal Society of London(53), 370-418.

Bebendorf, M. (2008). Hierarchical matrices. Springer.
Bengio, Y., Bengio, S., \& Cloutier, J. (1990). Learning a synaptic learning rule. Citeseer.

Blackwell, D., \& Dubins, L. (1962). Merging of opinions with increasing information. The Annals of Mathematical Statistics, 33(3), 882-886.

Blei, D. M., Kucukelbir, A., \& McAuliffe, J. D. (2017). Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518), 859-877.

Bostrom, N. (2014). Superintelligence: paths, dangers, strategies. Oxford University Press.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.
Brown, D., Niekum, S., \& Petrik, M. (2020). Bayesian robust optimization for imitation learning. Advances in Neural Information Processing Systems, 33, 2479-2491.

Brown, D. S., Cui, Y., \& Niekum, S. (2018). Risk-aware active inverse reinforcement learning. In Conference on robot learning (pp. 362-372).

Carey, R., Langlois, E., Everitt, T., \& Legg, S. (2020). The incentives that shape behaviour. arXiv preprint arXiv:2001.07118.

Chen, Z., \& Dongarra, J. J. (2005). Condition numbers of Gaussian random matrices. SIAM Journal on Matrix Analysis and Applications, 27(3), 603-620.

Christiano, P., Shlegeris, B., \& Amodei, D. (2018). Supervising strong learners by amplifying weak experts. arXiv preprint arXiv:1810.08575.

Christiano, P. F. (2014, Dec). Approval directed agents. Medium. Retrieved from https://ai-alignment.com/model-free-decisions $-6 e 6609 f 5 d 99 e$

Clouse, J. A. (1997). On integrating apprentice learning and reinforcement learning (Unpublished doctoral dissertation). University of Massachusetts Amherst.

Clouse, J. A., \& Utgoff, P. E. (1992). A teaching method for reinforcement learning. In Machine learning proceedings 1992 (pp. 92-101). Elsevier.

Cohen, M., Hutter, M., \& Osborne, M. (2022). Advanced artificial agents intervene in the provision of reward. AI Magazine, 43(3), 282-293.

Cohen, M. K., Catt, E., \& Hutter, M. (2019). A strongly asymptotically optimal agent in general environments. IJCAI.

Cohen, M. K., Catt, E., \& Hutter, M. (2021). Curiosity killed or incapacitated the cat and the asymptotically optimal agent. IEEE Journal on Selected Areas in Information Theory, 2(2), 665-677.

Cohen, M. K., Daulton, S., \& Osborne, M. A. (2022). Log-linear-time Gaussian processes using binary tree kernels. In Advances in neural information processing systems.

Cohen, M. K., \& Hutter, M. (2020). Pessimism about unknown unknowns inspires conservatism. In Conference on learning theory (pp. 1344-1373).

Cohen, M. K., Hutter, M., \& Nanda, N. (2022). Fully general online imitation learning. Journal of Machine Learning Research, 23(334), 1-30.

Cohen, M. K., Vellambi, B., \& Hutter, M. (2020). Asymptotically unambitious artificial general intelligence. In Proceedings of the AAAI conference on artificial intelligence.

Cutajar, K., Osborne, M., Cunningham, J., \& Filippone, M. (2016). Preconditioning kernel matrices. In International conference on machine learning (pp. 2529-2538).

Dietterich, T. G. (2000). Ensemble methods in machine learning. In Multiple classifier systems: First international workshop, mcs 2000 cagliari, italy, june

21-23, 2000 proceedings 1 (pp. 1-15).
Doob, J. L. (1953). Stochastic processes. John Wiley \& Sons.
Dorato, P. (1987). A historical review of robust control. IEEE Control Systems Magazine, 7(2), 44-47.

Dua, D., \& Graff, C. (2017). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml

Duchon, P., Flajolet, P., Louchard, G., \& Schaeffer, G. (2004, jul). Boltzmann samplers for the random generation of combinatorial structures. Comb. Probab. Comput., 13(4-5), 577-625. doi: 10.1017/S0963548304006315

Dullerud, G. E., \& Paganini, F. (2013). A course in robust control theory: a convex approach (Vol. 36). Springer Science \& Business Media.

Durrett, R. (2010). Probability: Theory and examples. Cambridge University Press.

Dutordoir, V., Durrande, N., \& Hensman, J. (2020). Sparse Gaussian processes with spherical harmonic features. In International conference on machine learning (pp. 2793-2802).

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics, 7(1), 1-26. Retrieved from https://doi.org/10.1214/ aos/1176344552 doi: 10.1214/aos/1176344552

Everitt, T., Filan, D., Daswani, M., \& Hutter, M. (2016). Self-modification of policy and utility function in rational agents. In International conference on artificial general intelligence (pp. 1-11).

Everitt, T., Hutter, M., Kumar, R., \& Krakovna, V. (2021). Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. Synthese, 1-33.

Feng, D., \& Baumgartner, R. (2020). Random forest (RF) kernel for regression, classification and survival. CoRR, abs/2009.00089. Retrieved from https: / / arxiv.org/abs/2009.00089

Fletcher, R. (2013). Practical methods of optimization (2nd ed.). New York: John Wiley \& Sons.

García, J., Acera, D., \& Fernández, F. (2013). Safe reinforcement learning through probabilistic policy reuse. RLDM 2013, 14.

García, J., \& Fernández, F. (2012). Safe exploration of state and action spaces in reinforcement learning. Journal of Artificial Intelligence Research, 45, 515564.

García, J., \& Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1), 1437-1480.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., \& Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration. Advances in neural information processing systems, 31.

Gaskett, C. (2003). Reinforcement learning under circumstances beyond its control. In Proceedings of the international conference on computational intelligence for modelling control and automation.

Gehring, C., \& Precup, D. (2013). Smart exploration in reinforcement learning using absolute temporal difference errors. In Proceedings of the 2013 international conference on autonomous agents and multi-agent systems (pp. 10371044).

Genewein, T., Delétang, G., Ruoss, A., Wenliang, L. K., Catt, E., Dutordoir, V., ... Veness, J. (2023, forthcoming). Memory-based meta-learning on nonstationary distributions. ICML.

Gilks, W. R., Richardson, S., \& Spiegelhalter, D. (1995). Markov chain monte Carlo in practice. CRC press.

Gold, E. M. (1967). Language identification in the limit. Information and control, 10(5), 447-474.

Hackbusch, W. (1999). A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. Computing, 62(2), 89-108.

Hackbusch, W., Khoromskij, B. N., \& Kriemann, R. (2004). Hierarchical matrices based on a weak admissibility criterion. Computing, 73(3), 207-243.

Hadfield-Menell, D., Dragan, A., Abbeel, P., \& Russell, S. (2017). The offswitch game. In Workshops at the thirty-first AAAI conference on artificial intelligence.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., \& Dragan, A. (2017). Inverse reward design. In Advances in neural information processing systems (pp. 6765-6774).

Hadfield-Menell, D., Russell, S. J., Abbeel, P., \& Dragan, A. (2016). Cooperative inverse reinforcement learning. In Advances in neural information processing systems (pp. 3909-3917).

Hans, A., Schneegaß, D., Schäfer, A. M., \& Udluft, S. (2008). Safe exploration for reinforcement learning. In Esann (pp. 143-148).

Hartikainen, J., \& Särkkä, S. (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In 2010 IEEE international
workshop on machine learning for signal processing (pp. 379-384).
Hastings, W. K. (1970). Monte Carlo sampling methods using markov chains and their applications. Biometrika, 57(1), 97.

Heger, M. (1994). Consideration of risk in reinforcement learning. In Machine learning proceedings 1994 (pp. 105-111). Elsevier.

Hensman, J., Durrande, N., \& Solin, A. (2017). Variational Fourier features for Gaussian processes. J. Mach. Learn. Res., 18(1), 5537-5588.

Hensman, J., Fusi, N., \& Lawrence, N. D. (2013). Gaussian processes for big data. In Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence (pp. 282-290). Arlington, Virginia, USA: AUAI Press.

Hensman, J., Rattray, M., \& Lawrence, N. (2012). Fast variational inference in the conjugate exponential family. Advances in neural information processing systems, 25.

Ho, J., \& Ermon, S. (2016). Generative adversarial imitation learning. In Advances in neural information processing systems (pp. 4565-4573).

Hoque, R., Balakrishna, A., Putterman, C., Luo, M., Brown, D. S., Seita, D., ... Goldberg, K. (2021). Lazydagger: Reducing context switching in interactive imitation learning. In 2021 IEEE 17th international conference on automation science and engineering (case) (pp. 502-509).

Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J., \& Garrabrant, S. (2019). Risks from learned optimization in advanced machine learning systems. arXiv preprint arXiv:1906.01820.

Hussein, A., Gaber, M. M., Elyan, E., \& Jayne, C. (2017). Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR), 50(2), 1-35.

Hutter, M. (2005). Universal artificial intelligence: Sequential decisions based on algorithmic probability. Berlin: Springer. doi: 10.1007/b138233

Hutter, M. (2009a). Discrete MDL predicts in total variation. In Advances in neural information processing systems 22 (NIPS'09) (pp. 817-825). Cambridge, MA, USA: Curran Associates.

Hutter, M. (2009b). Open problems in universal induction \& intelligence. Algorithms, 3(2), 879-906. doi: 10.3390/a2030879

Iyengar, G. N. (2005). Robust dynamic programming. Mathematics of Operations Research, 30(2), 257-280.

Jiang, G., Wu, C.-P., \& Cybenko, G. (1998). Minimax-based reinforcement learning with state aggregation. In Proceedings of the 37th IEEE conference on decision and control (cat. no. 98ch36171) (Vol. 2, pp. 1236-1241).

Judah, K., Fern, A. P., Dietterich, T. G., \& Tadepalli, P. (2014). Active imitation learning: Formal and practical reductions to iid learning. Journal of Machine Learning Research, 15(120), 4105-4143.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 83(1), 95-108.

Kalman, R. E. (1960). Contributions to the theory of optimal control. Bol. soc. mat. mexicana, 5(2), 102-119.

Kapoor, S., Finzi, M., Wang, K. A., \& Wilson, A. G. G. (2021, 18-24 Jul). SKIing on simplices: Kernel interpolation on the permutohedral lattice for scalable Gaussian processes. In M. Meila \& T. Zhang (Eds.), Proceedings of the 38th international conference on machine learning (Vol. 139, pp. 52795289). PMLR. Retrieved from https://proceedings.mlr.press/ v139/kapoor21a.html

Kirsch, L., Harrison, J., Sohl-Dickstein, J., \& Metz, L. (2022). Generalpurpose in-context learning by meta-learning transformers. arXiv preprint arXiv:2212.04458.

Koller, D., \& Friedman, N. (2009). Probabilistic graphical models: principles and techniques. MIT press.

Kosoy, V. (2019). Delegative reinforcement learning: learning to avoid traps with a little help. Safe Machine Learning workshop at ICLR.

Kraft, L. G. (1949). A device for quantizing, grouping, and coding amplitudemodulated pulses (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Kumar, R., Uesato, J., Ngo, R., Everitt, T., Krakovna, V., \& Legg, S. (2020). REALab: An embedded perspective on tampering. arXiv preprint arXiv:2011.08820.

Lakshminarayanan, B., Pritzel, A., \& Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems, 30.

Lang, T., Toussaint, M., \& Kersting, K. (2012). Exploration in relational domains for model-based reinforcement learning. The Journal of Machine Learning Research, 13(1), 3725-3768.

Lattimore, T., \& Hutter, M. (2011). Asymptotically optimal agents. In Proc. 22nd international conf. on algorithmic learning theory (ALT'11) (Vol. 6925, pp. 368-382). Espoo, Finland: Springer. doi: 10.1007/978-3-642-24412-4_29

Lattimore, T., \& Hutter, M. (2014a). Bayesian reinforcement learning with exploration. In International conference on algorithmic learning theory (pp. 170184).

Lattimore, T., \& Hutter, M. (2014b). General time consistent discounting. Theoretical Computer Science, 519, 140-154. doi: 10.1016/j.tcs.2013.09.022

Lattimore, T., \& Szepesvári, C. (2020). Bandit algorithms. Cambridge University Press.

Lázaro-Gredilla, M., \& Figueiras-Vidal, A. R. (2010). Marginalized neural network mixtures for large-scale regression. IEEE transactions on neural networks, 21(8), 1345-1351.

Lázaro-Gredilla, M., Quinonero-Candela, J., Rasmussen, C. E., \& FigueirasVidal, A. R. (2010). Sparse spectrum Gaussian process regression. The Journal of Machine Learning Research, 11, 1865-1881.

Leike, J. (2016). Nonparametric general reinforcement learning. arXiv preprint arXiv:1611.08944.

Leike, J., Lattimore, T., Orseau, L., \& Hutter, M. (2016). Thompson sampling is asymptotically optimal in general environments. In Proc. 32nd international conf. on uncertainty in artificial intelligence (UAI'16) (pp. 417-426). New Jersey, USA: AUAI Press.

Leike, J., Taylor, J., \& Fallenstein, B. (2016). A formal solution to the grain of truth problem. In Proceedings of the thirty-second conference on uncertainty in artificial intelligence (pp. 427-436).

Lévesque, J.-C., Durand, A., Gagné, C., \& Sabourin, R. (2017). Bayesian optimization for conditional hyperparameter spaces. In 2017 international joint conference on neural networks (ijcnn) (p. 286-293).

Li, L., Littman, M. L., \& Walsh, T. J. (2008). Knows what it knows: a framework for self-aware learning. In Proceedings of the 25th international conference on machine learning (pp. 568-575).

Li, M., \& Vitányi, P. (2008). An introduction to Kolmogorov complexity and its applications (Vol. 3). Springer.

Lu, X., Osband, I., Asghari, S. M., Gowal, S., Dwaracherla, V., Wen, Z., \& Van Roy, B. (2022). Robustness of epinets against distributional shifts. arXiv preprint arXiv:2207.00137.

Ma, X., \& Blaschko, M. (2020, 26-28 Aug). Additive tree-structured covariance function for conditional parameter spaces in Bayesian optimization. In S. Chiappa \& R. Calandra (Eds.), Proceedings of the twenty third international conference on artificial intelligence and statistics (Vol. 108, pp. 1015-1025). PMLR.

MacKay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge university press.

Maclin, R., \& Shavlik, J. W. (1996). Creating advice-taking reinforcement learners. Machine Learning, 22(1-3), 251-281.

Mcauliffe, J., \& Blei, D. (2007). Supervised topic models. Advances in neural information processing systems, 20.

Menda, K., Driggs-Campbell, K., \& Kochenderfer, M. J. (2019). Ensembledagger: A Bayesian approach to safe imitation learning. In 2019 IEEE/RSJ international conference on intelligent robots and systems (iros) (pp. 5041-5048).

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., \& Teller, E. (1953). Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6), 1087-1092.

Mhamdi, E. M. E., Guerraoui, R., Hendrikx, H., \& Maurer, A. (2017). Dynamic safe interruptibility for decentralized multi-agent reinforcement learning. In

Proceedings of the 31st international conference on neural information processing systems (pp. 129-139).

Milli, S., Hadfield-Menell, D., Dragan, A., \& Russell, S. (2017). Should robots be obedient? In Proceedings of the 26th international joint conference on artificial intelligence (pp. 4754-4760).

Moldovan, T. M., \& Abbeel, P. (2012). Safe exploration in Markov decision processes. In ICML.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., ... Jenatton, R. (2021). Uncertainty baselines: Benchmarks for uncertainty \& robustness in deep learning. arXiv preprint arXiv:2106.04015.

Nilim, A., \& El Ghaoui, L. (2005). Robust control of Markov decision processes with uncertain transition matrices. Operations Research, 53(5), 780-798.

Ober, S. W., \& Rasmussen, C. E. (2019). Benchmarking the neural linear model for regression. In Second symposium on advances in approximate Bayesian inference.

Olds, J. (1958). Self-stimulation of the brain: Its use to study local effects of hunger, sex, and drugs. Science, 127(3294), 315-324.

Omohundro, S. M. (2008). The basic AI drives. In Artificial general intelligence (Vol. 171, p. 483-492).

Orseau, L., \& Armstrong, S. (2016). Safely interruptible agents. In Proceedings of the thirty-second conference on uncertainty in artificial intelligence (pp. 557566).

Ortega, P. A., Wang, J. X., Rowland, M., Genewein, T., Kurth-Nelson, Z., Pascanu, R., ... Sprechmann, P. (2019). Meta-learning of sequential strategies. arXiv preprint arXiv:1905.03030.

Osband, I., Wen, Z., Asghari, S. M., Dwaracherla, V., Ibrahimi, M., Lu, X., \& Van Roy, B. (2021). Epistemic neural networks. arXiv preprint arXiv:2107.08924.

Pan, X., You, Y., Wang, Z., \& Lu, C. (2017). Virtual to real reinforcement learning for autonomous driving. arXiv preprint arXiv:1704.03952.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in neural information processing systems 32 (pp. 8024-8035). Curran Associates, Inc. Retrieved from http:// papers.neurips.cc/paper/9015-pytorch-an-imperative -style-high-performance-deep-learning-library.pdf

Poggio, T., Kur, G., \& Banburski, A. (2019). Double descent in the condition number. arXiv preprint arXiv:1912.06190.

Poland, J., \& Hutter, M. (2005). Asymptotics of discrete MDL for online prediction. IEEE Transactions on Information Theory, 51(11), 3780-3795. doi: 10.1109/TIT.2005.856956

Polymenakos, K., Laurenti, L., Patane, A., Calliess, J.-P., Cardelli, L., Kwiatkowska, M., . . Roberts, S. (2019). Safety guarantees for planning based on iterative Gaussian processes. arXiv preprint arXiv:1912.00071.

Psaros, A. F., Meng, X., Zou, Z., Guo, L., \& Karniadakis, G. E. (2023). Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. Journal of Computational Physics, 111902.

Quinonero-Candela, J., \& Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. The Journal of Machine Learning Research, 6, 1939-1959.

Rahimi, A., \& Recht, B. (2007). Random features for large-scale kernel machines. Advances in neural information processing systems, 20.

Riedl, M., \& Harrison, B. (2019). Enter the matrix: Safely interruptible autonomous systems via virtualization. In SafeAI@ AAAI.

Rigter, M., Lacerda, B., \& Hawes, N. (2022). RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, \& A. Oh (Eds.), Advances in neural information processing systems (Vol. 35, pp. 16082-16097). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/ 2022/file/6691c5e4a199b72dffd9c90acb63bcd6-Paper -Conference.pdf

Ring, M., \& Orseau, L. (2011). Delusion, survival, and intelligent agents. In Artificial general intelligence (pp. 11-20). Springer.

Ritchie, D., Horsfall, P., \& Goodman, N. D. (2016). Deep amortized inference for probabilistic programs. arXiv preprint arXiv:1610.05735.

Ross, S., \& Bagnell, D. (2010). Efficient reductions for imitation learning. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 661-668).

Ross, S., Gordon, G., \& Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the
fourteenth international conference on artificial intelligence and statistics (pp. 627-635).

Russell, S. (2019). Human compatible: Artificial intelligence and the problem of control. Penguin.

Samo, Y., \& Roberts, S. (2016). String and membrane Gaussian processes. Journal of Machine Learning Research, 17.

Särkkä, S. (2013). State space representation of Gaussian processes. Aalto University, Espoo, Finland. Pg, 22-23.

Särkkä, S., \& Hartikainen, J. (2012). Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In Artificial intelligence and statistics (pp. 993-1001).

Saunders, W., Sastry, G., Stuhlmueller, A., \& Evans, O. (2018). Trial without error: Towards safe reinforcement learning via human intervention. In Proceedings of the 17th international conference on autonomous agents and multiagent systems (pp. 2067-2069).

Schmidhuber, J., Zhao, J., \& Wiering, M. (1996). Simple principles of metalearning. Technical report IDSIA, 69, 1-23.

Schulte, P., \& Neander, K. (2022). Teleological Theories of Mental Content. In E. N. Zalta (Ed.), The Stanford encyclopedia of philosophy (Summer 2022 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2022/ entries/content-teleological/.

Simon, D. (2006). Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley \& Sons.

Snelson, E., \& Ghahramani, Z. (2005). Sparse Gaussian processes using pseudoinputs. In Y. Weiss, B. Schölkopf, \& J. Platt (Eds.), Advances in neural information processing systems (Vol. 18). MIT Press.

Solomonoff, R. J. (1964). A formal theory of inductive inference. Part I. Information and Control, 7(1), 1-22. doi: 10.1016/s0019-9958(64)90131-7

Sunehag, P., \& Hutter, M. (2015). Rationality, optimism and guarantees in general reinforcement learning. Journal of Machine Learning Research, 16, 13451390.

Syed, U., \& Schapire, R. E. (2010). A reduction from apprenticeship learning to classification. Advances in neural information processing systems, 23, 22532261.

Szita, I., \& Szepesvári, C. (2011). Agnostic KWIK learning and efficient approximate reinforcement learning. In Proceedings of the 24th annual conference on learning theory (pp. 739-772).

Talbott, W. (2016). Bayesian Epistemology. In E. N. Zalta (Ed.), The Stanford encyclopedia of philosophy (Winter 2016 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/ archives/win2016/entries/epistemology-bayesian/.

Tamar, A., Xu, H., \& Mannor, S. (2013). Scaling up robust MDPs by reinforcement learning. arXiv preprint arXiv:1306.6189.

Tannenbaum, A. (1980). Feedback stabilization of linear dynamical plants with uncertainty in the gain factor. International Journal of Control, 32(1), 1-16.

Taylor, J. (2016). Quantilizers: A safer alternative to maximizers for limited optimization. In AAAI workshop: AI, ethics, and society.

Taylor, J., Yudkowsky, E., LaVictoire, P., \& Critch, A. (2016). Alignment for advanced machine learning systems. Machine Intelligence Research Institute.

Thomaz, A. L., \& Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In Aaai (Vol. 6, pp. 1000-1005).

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3/4), 285-294.

Thrun, S., \& Pratt, L. (1998). Learning to learn: Introduction and overview. Learning to learn, 3-17.

Titsias, M. (2009a). Variational learning of inducing variables in sparse Gaussian processes. In Artificial intelligence and statistics (pp. 567-574).

Titsias, M. (2009b, 16-18 Apr). Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk \& M. Welling (Eds.), Proceedings of the twelth international conference on artificial intelligence and statistics (Vol. 5, pp. 567-574). Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR. Retrieved from https: //proceedings .mlr.press/v5/titsias09a.html

Turchetta, M., Berkenkamp, F., \& Krause, A. (2016). Safe exploration in finite Markov decision processes with Gaussian processes. Advances in Neural Information Processing Systems, 29.

Walsh, T., Goschin, S., \& Littman, M. (2010). Integrating sample-based planning and model-based reinforcement learning. In Proceedings of the aaai conference on artificial intelligence (Vol. 24, pp. 612-617).

Walsh, T. J., Szita, I., Diuk, C., \& Littman, M. L. (2009). Exploring compact reinforcement-learning representations with linear regression. In Proceedings
of the twenty-fifth conference on uncertainty in artificial intelligence (pp. 591598).

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., \& Wilson, A. G. (2019). Exact Gaussian processes on a million data points. Advances in Neural Information Processing Systems, 32.

Wen, Y., Tran, D., \& Ba, J. (2020). BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In International conference on learning representations. Retrieved from https://openreview.net/ forum?id=Sklf1yrYDr

Wenzel, F., Snoek, J., Tran, D., \& Jenatton, R. (2020). Hyperparameter ensembles for robustness and uncertainty quantification. Advances in Neural Information Processing Systems, 33, 6514-6527.

Wild, V., Kanagawa, M., \& Sejdinovic, D. (2021). Connections and equivalences between the Nyström method and sparse variational Gaussian processes. arXiv preprint arXiv:2106.01121.

Williams, C., \& Seeger, M. (2000). Using the Nyström method to speed up kernel machines. In T. Leen, T. Dietterich, \& V. Tresp (Eds.), Advances in neural information processing systems (Vol. 13). MIT Press. Retrieved from https://proceedings.neurips.cc/paper/2000/ file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf

Williams, C. K., \& Rasmussen, C. E. (2006). Gaussian processes for machine learning (Vol. 2) (No. 3). MIT press Cambridge, MA.

Wilson, A., \& Nickisch, H. (2015). Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In International conference on machine learning (pp. 1775-1784).

Yudkowsky, E. (2002). The AI-box experiment. Retrieved from http:// yudkowsky.net/singularity/aibox

Yudkowsky, E. (2008). Artificial intelligence as a positive and negative factor in global risk. Global catastrophic risks, l(303), 184.

Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. IEEE Transactions on automatic control, 26(2), 301-320.

Zhang, C., Bütepage, J., Kjellström, H., \& Mandt, S. (2018). Advances in variational inference. IEEE transactions on pattern analysis and machine intelligence, 41(8), 2008-2026.

Zhang, J., \& Cho, K. (2017). Query-efficient imitation learning for end-to-end simulated driving. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31).

Zhang, L., Zhou, W., \& Jiao, L. (2004). Wavelet support vector machine. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(1), 34-39.

Zhang, Y., Leithead, W. E., \& Leith, D. J. (2005). Time-series Gaussian process regression based on Toeplitz computation of $O\left(n^{2}\right)$ operations and $O(n)$-level storage. In Proceedings of the 44th IEEE conference on decision and control (pp. 3711-3716).

## A| Appendices

## A. 1 Definitions and Notation - Quick Reference

| Notation | Meaning |
| :--- | :--- |
| $\mathcal{A}, \mathcal{O}, \mathcal{R}$ | the finite action/observation/reward spaces |
| $\mathcal{H}$ | $\mathcal{A} \times \mathcal{O} \times \mathcal{R}$ |
| $h_{t}$ | $\in \mathcal{H} ;$ the interaction history in the $t^{\text {th }}$ timestep |
| $a_{t}, o_{t}, r_{t}$ | $\in \mathcal{A}, \mathcal{O}, \mathcal{R} ;$ the action, observation, and reward at timestep $t$ |
| $h_{<t}$ | $\left(h_{1}, \ldots, h_{t-1}\right)$ |
| $\nu, \mu$ | world-models stochastically mapping $\mathcal{H}^{*} \times \mathcal{A} \rightsquigarrow \mathcal{O} \times \mathcal{R}$ |
| $\mu$ | the true world-model/environment |
| $\mathcal{M}$ | the set of world-models the agent considers |
| $\pi$ | a policy stochastically mapping $\mathcal{H}^{*} \rightsquigarrow \mathcal{A}$ |
| $\mathrm{P}_{\nu}^{\pi}$ | a probability measure over histories with actions sampled from <br> $\pi$ and observations and rewards sampled from $\nu$ |
| $\mathbb{E}_{\nu}^{\pi}$ | the expectation when the interaction history is sampled from $\mathrm{P}_{\nu}^{\pi}$ |$|$| $\gamma$ | $\in[0,1) ;$ the agent's discount factor |
| :--- | :--- |
| $V_{\nu}^{\pi}\left(h_{<t}\right)$ | $(1-\gamma) \mathbb{E}_{\nu}^{\pi}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r_{k} \mid h_{<t}\right] ;$ the value of executing a policy $\pi$ <br> in an environment $\nu$ given the interaction history $h_{<t}$ |
| $\pi^{m}$ | the mentor's policy |
| $\mathcal{P}$ | the set of mentor-models the agent considers |
| $w(\nu)$ | the prior probability the agent assigns to $\nu$ being the true world- <br> model |


| $w^{\prime}(\pi)$ | the prior probability the agent assigns to $\pi$ being the mentor's policy |
| :---: | :---: |
| $w\left(\nu \mid h_{<t}\right)$ | the posterior probability that agent the assigns to $\nu$ after observing interaction history $h_{<t}$ |
| $w^{\prime}\left(\pi \mid h_{<t}\right)$ | the posterior probability that the agent assigns to the mentor's policy being $\pi$ after observing interaction history $h_{<t}$ |
| $\beta$ | $\in(0,1)$; tunes how pessimistic the agent is |
| $\mathcal{M}_{t}^{\beta}$ | top- $k$ world-models according $w\left(\cdot \mid h_{<t}\right)$, with $k$ chosen to satisfy $w\left(\mathcal{M}_{t}^{\beta} \mid h_{<t}\right)>\beta$ |
| $\pi^{\beta}\left(\cdot \mid h_{<t}\right)$ | $\left[\operatorname{argmax}_{\pi \in \Pi} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi}\left(h_{<t}\right)\right]\left(\cdot \mid h_{<t}\right)$ |
| $Z_{t}$ | positive i.i.d. random variable satisfying $p\left(Z_{t}<\varepsilon\right)>0$ and $p\left(Z_{t}>1\right)>0$ |
| $\theta_{t}$ | the probability the agent queries the mentor at time $t$ |
| $q_{t}$ | $\operatorname{Bern}\left(\theta_{t}\right)$; indicates whether the agent the queries mentor at time $t$ |
| $\pi_{Z}^{\beta}\left(\cdot \mid h_{<t}\right)$ | $\theta_{t} \pi^{m}\left(\cdot \mid h_{<t}\right)+\left(1-\theta_{t}\right) \pi^{\beta}\left(\cdot \mid h_{<t}\right)$ |
| $\mathcal{X}$ | general finite alphabet |
| $P, Q$ | probability measures over $\mathcal{X}^{\infty}$ |
| $x_{<t}$ | the first $t-1$ characters of $x_{<\infty} \in \mathcal{X}^{\infty}$ |
| $\omega, \Omega$ | $\omega$ is an outcome in a general sample space $\Omega$ |
| $d_{k}\left(P, Q \mid x_{<t}\right)$ | $k$-step variation distance $\max _{\mathcal{E} \subset \mathcal{X}^{k}}\left\|P\left(\mathcal{E} \mid x_{<t}\right)-Q\left(\mathcal{E} \mid x_{<t}\right)\right\|$ |
| $d\left(P, Q \mid x_{<t}\right)$ | total variation distance $\lim _{k \rightarrow \infty} d_{k}\left(P, Q \mid x_{<t}\right)$ |


| $\mathcal{F}, \mathcal{G}$ | sets of functions from $\mathbb{N}$ to $\mathbb{N}$ |
| :--- | :--- |
| $\mathrm{C}_{\mathcal{F G}}$ | $\operatorname{TIME}(\mathcal{F}) \cap \operatorname{SPACE}(\mathcal{G})$ |
| $\mathrm{FC}_{\mathcal{F G}}$ | a complexity class for environments $\nu$ (see Def. 4) |
| $E$ | $\subset \mathcal{H}^{*} \times \mathcal{A}$; an event |
| $E_{\leftarrow}$ | the set of interaction histories for which $E$ has happened <br> $\left\{h_{<t} a_{t} \in \mathcal{H}^{*} \times \mathcal{A}: \exists t^{\prime} \leq t h_{<t^{\prime}} a_{t^{\prime}} \in E\right\}$ |
| $c_{E}$ | a constant $>0$ depending on $E$ |
| Bayes $\mathcal{M}^{\prime}(\cdot)$ | for $\mathcal{M}^{\prime} \subset \mathcal{M},\left(\sum_{Q \in \mathcal{M}^{\prime}} w(Q) Q(\cdot)\right) / \sum_{Q \in \mathcal{M}^{\prime}} w(Q)$ |
| $V_{\nu}^{\pi \backslash k}\left(h_{<t}\right)$ | the truncated value $(1-\gamma) \mathbb{E}_{\nu}^{\pi}\left[\sum_{j=t}^{t+k-1} \gamma^{j-t} r_{j} \mid h_{<t}\right]$ |
| $\lim$ | $\lim _{t \rightarrow \infty}$ |
| $w . p .1$ | with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$ probability 1 |

## A. 2 Algorithm for Pessimism

$\pi^{\beta}$ is defined to optimize the pessimistic value, but for this algorithm, $\pi^{\beta}$ picks an action that is $\varepsilon$-optimal, as is necessary for infinite-horizon planning. Algorithm 4 takes a set of world-models or mentor-models $\mathcal{M}=\left\{\nu_{i}\right\}_{i \in \mathbb{N}}$ or $\left\{\pi_{i}\right\}_{i \in \mathbb{N}}$, a prior $w$, a threshold $\alpha$, and a history $h_{<t}$. It calculates the posterior $w\left(\cdot \mid h_{<t}\right)$ to enough precision, for enough models, to identify a minimal set $\mathcal{M}_{t}^{\alpha} \subset \mathcal{M}$ such that $w\left(\mathcal{M}_{t}^{\alpha} \mid h_{<t}\right)>\alpha$. It returns $\mathcal{M}_{t}^{\alpha}$, and the last model added to $\mathcal{M}_{t}^{\alpha}$. $\mathcal{M}$ must be ordered so that $i<j \Longrightarrow w\left(\nu_{i}\right) \geq w\left(\nu_{j}\right)$.

Algorithm 5 samples from the $\varepsilon$-optimal version of $\pi_{Z}^{\beta}$.

## A. 3 Proofs of Lemmas

Lemma 1 (Merging of Top Opinions). For $\beta \in(0,1), \lim _{t \rightarrow \infty} \max _{Q \in \mathcal{M}_{t}^{\beta}} d\left(P, Q \mid x_{<t}\right)=$ 0 with P-probability 1 (i.e. when $x_{<\infty}=\omega \sim P$ ).


#### Abstract

Algorithm 4 Calculate Posterior Up to Threshold. The posterior cannot be computed exactly, since the normalization constant is an infinite sum. It suffices for our purposes to compute it to finite precision. This complication makes the algorithm more involved, so unless the reader is particularly interested or skeptical, the details of this algorithm are non-essential.


Require: $\mathcal{M}=\left\{\rho_{i}\right\}_{i \in \mathbb{N}}, w: \mathcal{M} \rightarrow[0,1], \alpha, h_{<t}$, where $i<j \Longrightarrow w\left(\rho_{i}\right) \geq$ $w\left(\rho_{j}\right)$
$W \leftarrow$ [empty list] $\triangleright$ contains un-normalized posterior weights
$\Sigma_{W} \leftarrow 0 \quad \triangleright$ sum of $W$
$\Sigma_{*} \leftarrow 1 \quad \triangleright$ sum of prior weights of unchecked $\rho_{i}$
$i \leftarrow 1 \quad \triangleright$ index of first unchecked $\rho_{i}$
while True do
$W[i] \leftarrow w\left(\rho_{i}\right)$
$\Sigma_{*} \leftarrow \Sigma_{*}-W[i]$
for $k \leftarrow 0$ to $t-1$ do
$W[i] \leftarrow W[i] *\left[\rho_{i}\left(o_{k}, r_{k} \mid h_{<k} a_{k}\right)\right.$ or $\left.\rho_{i}\left(a_{k} \mid h_{<k}\right)\right]$ (depending on whether $\rho$ is world-model or mentor-model)
$\Sigma_{W} \leftarrow \Sigma_{W}+W[i]$
cutoff $\leftarrow w\left(\rho_{i+1}\right) \quad \triangleright$ for a checked world-model
to definitely $\in \mathcal{M}_{t}^{\beta}$, its un-normalized posterior weight must be at least cutoff; otherwise, the first unchecked model might have larger posterior weight
$J \leftarrow[1,2, \ldots, i]$
sort $J$ by $W$ descending
weight_sum $\leftarrow 0$
last_added $\leftarrow$ null
$\mathcal{M}_{t}^{\alpha} \leftarrow \emptyset$
last_model $\leftarrow$ null
for $j \in J$ do
if $W[j]<$ cutoff then break
weight_sum $\leftarrow$ weight_sum $+W[j]$
last_added $\leftarrow W[j]$
last_model $\leftarrow \rho_{j}$
$\mathcal{M}_{t}^{\alpha} \leftarrow \mathcal{M}_{t}^{\alpha} \cup\left\{\rho_{j}\right\}$
$\triangleright$ Note $\Sigma_{W} \leq \sum_{\rho \in \mathcal{M}}[$ un-normalized posterior weight of $\rho] \leq \Sigma_{W}+\Sigma_{*}$, so $w\left(\mathcal{M}_{t}^{\alpha} \mid h_{<t}\right) \geq \frac{\text { weight_sum }}{\Sigma_{W}+\Sigma_{*}}$ and $w\left(\mathcal{M}_{t}^{\alpha} \backslash \rho_{j} \mid h_{<t}\right) \leq \frac{\text { weight_sum-last_added }}{\Sigma_{W}}$
if $\frac{\text { weight_sum }}{\Sigma_{W}+\Sigma_{*}}>\alpha$ then $\quad \triangleright$ these models cover $>\alpha$ of posterior
if $\frac{\text { weight_sum-last_added }}{\Sigma_{W}} \leq \alpha$ then $\quad \triangleright$ the last one is definitely needed
return $\mathcal{M}_{t}^{\alpha}$, last_model
break
$i \leftarrow i+1$

```
Algorithm \(5 \varepsilon\)-optimal approximation of \(\pi_{Z}^{\beta}\left(\cdot \mid h_{<t}\right)\). The agent does a variant of
expectimax planning, in which a minimum over \(\nu \in \mathcal{M}_{t}^{\beta}\) appears at each step. Then it uses a Thompson sampling-inspired approach to decide whether to query the mentor.
Require: \(\mathcal{A}, \mathcal{O}, \mathcal{R}, \mathcal{M}=\left\{\nu_{i}\right\}_{i \in \mathbb{N}}, w: \mathcal{M} \rightarrow[0,1], \mathcal{P}=\left\{\pi_{i}\right\}_{i \in \mathbb{N}}, w^{\prime}: \mathcal{P} \rightarrow\) \([0,1], \gamma, \beta, \operatorname{Dist}(Z), h_{<t}, \varepsilon\)
\(k \leftarrow\left\lceil\log _{\gamma}(\varepsilon)\right\rceil \quad \triangleright\) the agent need only consider a horizon of \(k\) to estimate the value within \(\varepsilon\)
\(\mathcal{H} \leftarrow \mathcal{A} \times \mathcal{O} \times \mathcal{R}\)
\(\mathcal{M}_{t}^{\beta},_{-} \leftarrow\) Calculate Posterior Up to \(\operatorname{Threshold}\left(\mathcal{M}, w, \beta, h_{<t}\right)\)
for \(h^{k} \in \mathcal{H}^{k}\) do \(V_{h^{k}} \leftarrow(1-\gamma) \sum_{j=0}^{k-1} \gamma^{j} r_{j}^{k}\) (where \(a_{j}^{k}, o_{j}^{k}\), and \(r_{j}^{k}\) are the \(j^{\text {th }}\) action, observation, and reward of \(h^{k}\) )
for \(j \leftarrow k-1\) to 0 do
\[
\begin{aligned}
& \text { for } h^{j} \in \mathcal{H}^{j} \text { do } \quad \triangleright \text { note } \mathcal{H}^{0}=\{\emptyset\} \\
& V_{h^{j}} \leftarrow \max _{a \in \mathcal{A}} \min _{\nu \in \mathcal{M}_{t}^{\beta}} \sum_{o, r \in \mathcal{O} \times \mathcal{R}} \nu\left(o, r \mid h_{<t} h^{j} a\right) V_{h^{j} \text { aor }}
\end{aligned}
\]
\(Y_{t} \leftarrow V_{\emptyset}\)
\(a_{t}^{\beta} \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \min _{\nu \in \mathcal{M}_{t}^{\beta}} \sum_{o, r \in \mathcal{O} \times \mathcal{R}} \nu\left(o, r \mid h_{<t} a\right) V_{\text {aor }}\)
if \(Y_{t}=0\) then return query mentor
\(\theta_{1}, \theta_{2} \sim \operatorname{Uniform}(0,1)\)
\({ }_{-}, \pi \leftarrow\) Calculate Posterior Up to Threshold \(\left(\mathcal{P}, w^{\prime}, \theta_{1}, h_{<t}\right)\)
_, \(\nu \leftarrow\) Calculate Posterior Up to \(\operatorname{Threshold}\left(\mathcal{M}, w, \theta_{2}, h_{<t}\right)\)
\(X_{t} \leftarrow \sum_{h^{k} \in \mathcal{H}^{k}}\left[\prod_{j=0}^{k-1} \pi\left(a_{j}^{k} \mid h_{<t} h_{<j}^{k}\right) \nu\left(o_{j}^{k} r_{j}^{k} \mid h_{<t} h_{<j}^{k} a_{j}^{k}\right)\right](1-\gamma) \sum_{j=0}^{k-1} \gamma^{j} r_{j}^{k}\)
\(Z_{t} \sim \operatorname{Dist}(Z)\)
if \(X_{t}>Y_{t}+Z_{t}\) then return query mentor else return \(a_{t}^{\beta}\)
```

We convert the Merging of Top Opinions Lemma into an on-policy learning result for the pessimistic agent.

Corollary 5 (On-Policy Prediction).

$$
\lim \max _{\nu \in \mathcal{M}_{t}^{\beta}} d\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)=0 \text { w.p.l }
$$

Proof. We convert the problem to a sequence prediction problem as follows. Let $\widetilde{\mathcal{M}}=\left\{\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}} \mid \nu \in \mathcal{M}\right\}$, and let $\widetilde{w}\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}\right)=w(\nu)$. For any history with positive $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$ probability, $\widetilde{w}\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)=w\left(\nu \mid h_{<t}\right)$, so $\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}} \in \widetilde{\mathcal{M}}_{t}^{\beta}$ if and only if $\nu \in \mathcal{M}_{t}^{\beta}$. Therefore,

$$
\lim \max _{\nu \in \mathcal{M}_{t}^{\beta}} d\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)=\lim \max _{\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}} \in \widetilde{\mathcal{M}}_{t}^{\beta}} d\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)=0 \text { w.p. } 1
$$

by Lemma 1 (the Merging of Top Opinions Lemma).

We will make use of the "truncated value", defined as follows:

$$
\begin{equation*}
V_{\nu}^{\pi \backslash k}\left(h_{<t}\right):=(1-\gamma) \mathbb{E}_{\nu}^{\pi}\left[\sum_{j=t}^{t+k-1} \gamma^{j-t} r_{j} \mid h_{<t}\right] \tag{A.1}
\end{equation*}
$$

We will often consider the truncated value while exploiting the fact that

$$
\begin{equation*}
0 \leq V_{\nu}^{\pi}\left(h_{<t}\right)-V_{\nu}^{\pi \backslash k}\left(h_{<t}\right) \leq \gamma^{k} \tag{A.2}
\end{equation*}
$$

which follows from $r_{j} \in[0,1]$.
The following lemma is an intermediate result in the proof of Leike, Lattimore, et al.'s (2016) Lemma 2, and the proof is transcribed with notational changes.

Lemma 10 (Variation Distance Bounds Expectation-Difference). Let $P_{1}$ and $P_{2}$
be two probability measures defined on the same space, and let $X \in[0,1]$ be a random variable. Then

$$
\left|\mathbb{E}_{P_{1}}[X]-\mathbb{E}_{P_{2}}[X]\right| \leq d\left(P_{1}, P_{2}\right)
$$

Proof. Let $Q=\left(P_{1}+P_{2}\right) / 2$. Let $\frac{d P_{i}}{d Q}(\omega)$ denote the Radon Nykodym-derviative, where $\omega \in \Omega$ is a generic outcome. Let $A$ be the event $\frac{d P_{1}}{d Q}(\omega) \geq \frac{d P_{2}}{d Q}(\omega)$ Then

$$
\begin{aligned}
\mathbb{E}_{P_{1}}[X]-\mathbb{E}_{P_{2}}[X] & =\mathbb{E}_{\omega \sim Q}\left[X(\omega) \frac{d P_{1}}{d Q}(\omega)-X(\omega) \frac{d P_{2}}{d Q}(\omega)\right] \\
& \leq \mathbb{E}_{\omega \sim Q}\left[\left.X(\omega)\left(\frac{d P_{1}}{d Q}(\omega)-\frac{d P_{2}}{d Q}(\omega)\right) \right\rvert\, \omega \in A\right] \\
& \leq \mathbb{E}_{\omega \sim Q}\left[\left.\frac{d P_{1}}{d Q}(\omega)-\frac{d P_{2}}{d Q}(\omega) \right\rvert\, \omega \in A\right] \\
& =P_{1}(A)-P_{2}(A) \leq \sup _{A \in \mathcal{F}}\left|P_{1}(A)-P_{2}(A)\right|=d\left(P_{1}, P_{2}\right)
\end{aligned}
$$

Since variation distance is symmetric, $\left|\mathbb{E}_{P_{1}}[X]-\mathbb{E}_{P_{2}}[X]\right| \leq d\left(P_{1}, P_{2}\right)$.

The following is a simple consequence.
Lemma 11. $\left|V_{\nu}^{\pi}\left(h_{<t}\right)-V_{\mu}^{\pi}\left(h_{<t}\right)\right|>\varepsilon>0 \Longrightarrow d_{\left\lceil\log _{\gamma}(\varepsilon / 2)\right\rceil}\left(\mathrm{P}_{\nu}^{\pi}, \mathrm{P}_{\mu}^{\pi} \mid h_{<t}\right)>$ $\varepsilon / 2>0$

Proof. Letting $k=\left\lceil\log _{\gamma}(\varepsilon / 2)\right\rceil,\left|V_{\nu}^{\pi}\left(h_{<t}\right)-V_{\mu}^{\pi}\left(h_{<t}\right)\right|>\varepsilon$ implies $\mid V_{\nu}^{\pi \backslash k}\left(h_{<t}\right)-$ $V_{\mu}^{\pi \backslash k}\left(h_{<t}\right) \mid>\varepsilon / 2$ by Inequality A.2. Since the value is bounded by [0, 1], from Lemma 10,

$$
\begin{equation*}
\left|V_{\nu}^{\pi \backslash k}\left(h_{<t}\right)-V_{\mu}^{\pi \backslash k}\left(h_{<t}\right)\right| \leq d_{k}\left(\mathrm{P}_{\nu}^{\pi}, \mathrm{P}_{\mu}^{\pi} \mid h_{<t}\right) \tag{A.3}
\end{equation*}
$$

so $d_{\left\lceil\log _{\gamma}(\varepsilon / 2)\right\rceil}\left(\mathrm{P}_{\nu}^{\pi}, \mathrm{P}_{\mu}^{\pi} \mid h_{<t}\right)>\varepsilon / 2>0$.
Corollary 6 (Finite Zero Conditions). The zero condition, in which the agent queries the mentor because the pessimistic value of all policies is 0 , only occurs
finitely often, with probability 1.
Proof. By the previous two lemmas, the pessimistic value of $\pi_{Z}^{\beta}$ approaches the true value with probability 1 , and the true value is at least $\varepsilon_{r}$ because rewards less than $\varepsilon_{r}$ are never provided. Thus, eventually, there is always at least one policy with a pessimistic value greater than 0 , so the zero condition is never met thereafter.

Since all our remaining performance results consider limiting behavior, we will ignore the zero condition.

The next lemma, from M. K. Cohen et al. (2020, Lemma 3), states that the posterior probability on the truth (regarding both the true world-model and the true mentor-model) does not approach 0 .

Lemma 12 (Posterior on Truth).

$$
P\left[\inf _{t} w\left(P \mid x_{<t}\right)=0\right]=0
$$

Proof. If $w\left(P \mid x_{<t}\right)=0$ for some $t$, then $P\left(x_{<t}\right)=0$, so with $P$-probability $1, \inf _{t \in \mathbb{N}} w\left(P \mid x_{<t}\right)=0 \Longrightarrow \liminf _{t \in \mathbb{N}} w\left(P \mid x_{<t}\right)=0$ which in turn implies $\lim \sup _{t \in \mathbb{N}} w\left(P \mid x_{<t}\right)^{-1}=\infty$. We show that this has probability 0 .

Let $z_{t}:=w\left(P \mid x_{<t}\right)^{-1}$. We show that $z_{t}$ is a $P$-martingale.

$$
\begin{align*}
\mathbb{E}_{P}\left[z_{t+1} \mid x_{<t}\right] & \stackrel{(a)}{=} \mathbb{E}_{P}\left[w\left(P \mid x_{t+1}\right)^{-1} \mid x_{<t}\right] \\
& \stackrel{(b)}{=} \sum_{\bar{x} \in \mathcal{X}} P\left(\bar{x} \mid x_{<t}\right)\left[\frac{\operatorname{BayesM}\left(x_{t} \bar{x}\right)}{w(P) P\left(x_{t} \bar{x}\right)}\right] \\
& \stackrel{(c)}{=} \sum_{\bar{x} \in \mathcal{X}} \frac{\operatorname{BayesM}\left(x_{t} \bar{x}\right)}{w(P) P\left(x_{<t}\right)} \\
& \stackrel{(d)}{=} \sum_{\bar{x} \in \mathcal{X}} \operatorname{BayesM}\left(\bar{x} \mid x_{t}\right) \frac{\operatorname{Bayes} \mathcal{M}\left(x_{t}\right)}{w(P) P\left(x_{<t}\right)} \\
& \stackrel{(())}{=} \frac{\operatorname{BayesM}\left(x_{t}\right)}{w(P) P\left(x_{<t}\right)} \\
& \stackrel{(f)}{=} w\left(P \mid x_{<t}\right)^{-1} \\
& =z_{t} \tag{A.4}
\end{align*}
$$

where (a) is the definition of $z_{t}$, (b) follows from Bayes' Rule, (c) follows from multiplying the numerator and denominator by Bayes $\mathcal{M}\left(x_{<t}\right)$ and cancelling, (d) follows from expanding the numerator, (e) follows because Bayes $\mathcal{M}$ is a measure, and (f) follows from Bayes' Rule, completing the proof that $z_{t}$ is martingale. By the martingale convergence theorem $z_{t} \rightarrow f(\omega)<\infty$ w.p.1, for $\omega \in \Omega$, the sample space, and some $f: \Omega \rightarrow \mathbb{R}$, so the probability that $\limsup _{i \in \mathbb{N}} w\left(P \mid x_{<t}\right)^{-1}=$ $\infty$ is 0 , completing the proof.

Note that the posterior probability on the mentor-policy is only updated at some timesteps (when the mentor is queried), but it is clearly still a martingale.

Lemma 6 (Almost On-Policy Convergence). For a sequence of policies $\pi_{t}$ and an infinite set of timesteps $\tau$, the following holds with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$-prob. 1: if there exists $c>0$ such that $\forall t \in \tau \forall t^{\prime} \geq t \forall a \in \mathcal{A} \pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq c \pi_{t}\left(a \mid h_{<t^{\prime}}\right)$, then $\lim _{\tau \ni t \rightarrow \infty} V_{\mu}^{\pi_{t}}\left(h_{<t}\right)-\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi_{t}}\left(h_{<t}\right)=0$ and for all $k, \lim _{\tau \ni t \rightarrow \infty} \max _{\nu \in \mathcal{M}_{t}^{\beta}}$
$d_{k}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)=0$.
Proof. Suppose by contradiction that $\left|\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi_{t}}\left(h_{<t}\right)-V_{\mu}^{\pi_{t}}\left(h_{<t}\right)\right|>\varepsilon>$ 0 infinitely often for $t \in \tau$. Then, by Lemma 11 , for some $\nu \in \mathcal{M}_{t}^{\beta}$ at each of those timesteps, $d_{\left[\log _{\gamma}(\varepsilon / 2)\right\rceil}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)>\varepsilon / 2>0$. So then there exists a $k$ for which $\max _{\nu \in \mathcal{M}_{t}^{\beta}} d_{k}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)>\varepsilon / 2>0$ infinitely often for $t \in$ $\tau$. Now we are supposing a contradiction in either of the two implications of the theorem. An event on which the two measures differ by at least $\varepsilon / 2$ occurs within $k$ timesteps. Because $\pi_{Z}^{\beta}\left(\cdot \mid h_{<t^{\prime}}\right) \geq c \pi_{t}\left(\cdot \mid h_{<t^{\prime}}\right), d_{k}\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right) \geq$ $c^{k} d_{k}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)$. This holds for any $\nu$, but in particular for $\nu \in \mathcal{M}_{t}^{\beta}$, so $\max _{\nu \in \mathcal{M}_{t}^{\beta}} d_{k}\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right) \geq c^{k} \max _{\nu \in \mathcal{M}_{t}^{\beta}} d_{k}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)>c^{k} \varepsilon / 2$. This happens infinitely often for $t \in \tau$.
$\operatorname{But} d\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right) \geq d_{k}\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)$, so $\max _{\nu \in \mathcal{M}_{t}^{\beta}} d\left(\mathrm{P}_{\nu}^{\pi_{Z}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right)>$ $c^{k} \varepsilon / 2>0$ infinitely often, which has probability 0 by Corollary 5 . Thus, the original assumption has probability 0 , completing the proof.

We complete the proof of Theorem 1 here.

Proof. (Theorem 1) The proof begins in the main paper, in a "detailed proof outline". Recall the inductive hypotheses:

- $t_{k}$ exists: a timestep after which
$-\max _{\nu \in \mathcal{M}_{t}^{\alpha}}\left|V_{\nu}^{\pi^{\prime} k ; \pi^{\beta}}\left(h_{<t}\right)-V_{\mu}^{\pi^{\prime} k ; \pi^{\beta}}\left(h_{<t}\right)\right|<\varepsilon$
$-\max _{\nu \in \mathcal{M}_{t}^{\alpha}} d_{k}\left(\mathrm{P}_{\nu}^{\pi^{\prime}}, \mathrm{P}_{\mu}^{\pi^{\prime}} \mid h_{<t}\right)<\varepsilon$
for all $t \in \tau_{k-1}$
- $\left|\tau_{k}\right|=\infty$, where $t \in \tau_{k}$ if and only if
- $t \in \tau_{k-1}$ (and for $\tau_{0}, t \in \tau^{\times}$as well)

$$
\begin{aligned}
& \text { - } t \geq t_{k} \\
& \text { - } \forall t^{\prime}<k: \theta_{t+t^{\prime}} \geq \nu_{\mathrm{inf}}^{\prime} \pi_{\mathrm{inf}}^{\prime} p\left(Z_{t+t^{\prime}}<\varepsilon\right) \\
& -V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)+2 \varepsilon
\end{aligned}
$$

The proof by induction starts with $k=0 . \tau_{-1}=\mathbb{N}$, so $t_{0}$ is a timestep after which $\max _{\nu \in \mathcal{M}_{t}^{\alpha}}\left|V_{\nu}^{\pi^{\beta}}-V_{\mu}^{\pi^{\beta}}\right|<\varepsilon$ for all $t \geq t_{0}$. From Lemma 6, setting $\pi_{t}=\pi^{\beta}$, setting $\tau=\tau_{-1}$, setting $\beta^{\prime}=\alpha$, and setting $c=p\left(Z_{t}>1\right)>0$, the condition of the lemma holds-that $\forall t \in \tau \forall t^{\prime} \geq t, \pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq c \pi_{t}\left(a \mid h_{<t^{\prime}}\right) \forall a \in \mathcal{A}$ so we have the result that with probability $1, \lim _{\mathbb{N} \ni t \rightarrow \infty} \max _{\nu \in \mathcal{M}_{t}^{\alpha}} \mid V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right)-$ $V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right) \mid=0$. Therefore, $t_{0}$ exists with probability 1 . Turning to $\tau_{0}$, the first and the third condition are immediate, so we need only show that the fourth condition is satisfied infinitely often with probability 1 for $t \in \tau^{\times}$, namely that $V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t}\right) \geq$ $V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right)+2 \varepsilon$. This is true for all $t \in \tau^{\times}$, and $\left|\tau^{\times}\right|=\infty$.

Now we show that if $t_{k}$ exists and $\left|\tau_{k}\right|=\infty$, then with probability $1, t_{k+1}$ exists and $\left|\tau_{k+1}\right|=\infty$. For each $t \in \tau_{k}, V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)+2 \varepsilon$. For $t>t_{0}, \max _{\nu \in \mathcal{M}_{t}^{\alpha}}\left|V_{\nu}^{\pi^{\beta}}\left(h_{<t+k}\right)-V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)\right|<\varepsilon$, and since $\alpha \geq \beta$, $\mathcal{M}_{t}^{\beta} \subset \mathcal{M}_{t}^{\alpha}$, so $\max _{\nu \in \mathcal{M}_{t}^{\beta}}\left|V_{\nu}^{\pi^{\beta}}\left(h_{<t+k}\right)-V_{\mu}^{\pi^{\beta}}\left(h_{<t+k}\right)\right|<\varepsilon$. Combining these, we have $V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k}\right) \geq \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t+k}\right)+\varepsilon$ for $t \in \tau_{k}$. Thus, the probability of exploring $\theta_{t+k} \geq \nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p\left(Z_{t+k}<\varepsilon\right)>0$. Since $A(t, k)$ holds for $t \in \tau_{k}$, $A(t, k+1)$ holds as well.

In preparation to apply Lemma 6 , let $\pi_{t}=\left(\pi^{\prime}(k+1) ; \pi^{\beta}\right)_{t}$; that is, since $\pi_{t}$ need only be defined from timestep $t$ onward, let $\pi_{t}$ be the policy which follows $\pi^{\prime}$ from timestep $t$ through timestep $t+k$, and follows $\pi^{\beta}$ thereafter. Set $\tau$ from Lemma 6 to be $\tau_{k}$. For $t^{\prime}>t+k, \pi_{t}\left(\cdot \mid h_{<t^{\prime}}\right)=\pi^{\beta}\left(\cdot \mid h_{<t^{\prime}}\right)$, which satisfies $\pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq$ $c \pi^{\beta}\left(a \mid h_{<t^{\prime}}\right) \forall a \in \mathcal{A}$. For $t \leq t^{\prime} \leq t+k, \theta_{t^{\prime}} \geq \nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p(Z<\varepsilon)$, this being the proposition $A(t, k+1)$. Since $\pi_{Z}^{\beta}$ mimics the mentor's policy $\pi^{m}$ when exploring, for $t \leq t^{\prime} \leq t+k, \pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq c \pi^{m}\left(a \mid h_{<t^{\prime}}\right) \forall a \in \mathcal{A}$, for $c=\nu_{\mathrm{inf}}^{\prime} \pi_{\mathrm{inf}}^{\prime} p(Z<\varepsilon)$.

But we need that $\pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq c^{\prime} \pi^{\prime}\left(a \mid h_{<t^{\prime}}\right) \forall a \in \mathcal{A}$.
So we show that $d_{1}\left(\pi^{\prime}, \pi^{m} \mid h_{<t}\right) \theta_{t} \rightarrow 0$ with probability 1 . For a mentor-model $\pi_{i} \in \mathcal{P}$, consider the alternative policy to $\pi_{Z}^{\beta}$, which explores by mimicking $\pi_{i}$ instead of $\pi^{m}$. Call this policy $\pi_{Z, i}^{\beta}$ Consider a prior over probability measures where $w^{\prime \prime}\left(\mathrm{P}_{\mu}^{\pi_{z, i}^{\beta}}\right):=w^{\prime}\left(\pi_{i}\right)$, and note that $w^{\prime \prime}\left(\mathrm{P}_{\mu}^{\pi_{z, i}^{\beta}} \mid h_{<t}\right)=w^{\prime}\left(\pi_{i} \mid h_{<t}\right)$. Because $w^{\prime}\left(\pi^{\prime} \mid h_{<t}\right) \geq \pi_{\text {inf }}^{\prime}, w^{\prime \prime}\left(\mathrm{P}_{\mu}^{\pi_{z,}^{\beta}} \mid h_{<t}\right) \geq \pi_{\text {inf }}^{\prime}$. By Lemma 3, this implies $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[d\left(\mathrm{P}_{\mu}^{\pi_{z,}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right) \rightarrow 0\right]=1$. Trivially, $d\left(\mathrm{P}_{\mu}^{\pi_{z, \prime}^{\beta}}, \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} \mid h_{<t}\right) \geq d_{1}\left(\pi^{\prime}, \pi^{m} \mid h_{<t}\right) \theta_{t}$, so $d_{1}\left(\pi^{\prime}, \pi^{m} \mid h_{<t}\right) \theta_{t} \rightarrow 0$ with probability 1 .

Recall that for $t \leq t^{\prime} \leq t+k, \theta_{t^{\prime}}$ is uniformly bounded below, so on those timesteps, $d_{1}\left(\pi^{\prime}, \pi^{m} \mid h_{<t}\right) \rightarrow 0$. Therefore, there exists a time $t_{k}^{\prime}$ after which $\pi^{m}\left(a \mid h_{<t^{\prime}}\right) \geq \pi^{\prime}\left(a \mid h_{<t^{\prime}}\right) / 2 \forall a \in \mathcal{A}$. This gives us that for those timesteps $t \leq$ $t^{\prime} \leq t+k$, for $t \in \tau_{k}$ and $\geq t_{k}^{\prime}$, for all $a \in \mathcal{A}$,

$$
\begin{equation*}
\pi_{Z}^{\beta}\left(a \mid h_{<t^{\prime}}\right) \geq \nu_{\mathrm{inf}}^{\prime} \pi_{\mathrm{inf}}^{\prime} p(Z<\varepsilon) / 2 \pi^{\prime}\left(a \mid h_{<t^{\prime}}\right) \tag{A.5}
\end{equation*}
$$

Restricting $\tau$ to be the set of timesteps in $\tau_{k}$ after $t_{k}^{\prime}, \tau$ is still infinite, and we can now apply Lemma 6 on the policy $\pi_{t}=\left(\pi^{\prime}(k+1) ; \pi^{\beta}\right)_{t}$, with $\beta^{\prime}=\alpha$ again, and with $c=\nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p(Z<\varepsilon) / 2$. The implication of the lemma is that $\lim _{\tau_{k} \ni t \rightarrow \infty} \max _{\nu \in \mathcal{M}_{t}^{\alpha}}\left|V_{\nu}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)-V_{\mu}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)\right|=0$ and for all $j$, $\lim _{\tau \ni t \rightarrow \infty} \max _{\nu \in \mathcal{M}_{t}^{\beta}} d_{j}\left(\mathrm{P}_{\nu}^{\pi_{t}}, \mathrm{P}_{\mu}^{\pi_{t}} \mid h_{<t}\right)=0$. In particular, this holds for $j=k+1$. Together, these imply that $t_{k+1}$, a time after which the value difference and the variation distance are both less than $\varepsilon$, exists. (For the $k+1$-step variation distance, $\pi_{t}$ is equivalent to $\pi^{\prime}$ ).

Since $\left|\tau_{k}\right|=\infty$, we have already shown that the first three conditions are satisfied infinitely often. So to show that $\left|\tau_{k+1}\right|=\infty$, we need only show that among those infinitely many timesteps, the following condition holds infinitely often:
$V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right) \geq V_{\mu}^{\pi^{\beta}}\left(h_{<t+k+1}\right)+2 \varepsilon$. We begin,

$$
\begin{align*}
V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t}\right) & \stackrel{(a)}{\geq} V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right)+7 \varepsilon \varepsilon^{(b)} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right)+6 \varepsilon \stackrel{(c)}{\geq} \min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)+6 \varepsilon \\
& \stackrel{(d)}{\geq} V_{\mu}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)+5 \varepsilon \stackrel{(e)}{\geq} V_{\nu^{\prime}}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)+4 \varepsilon \tag{A.6}
\end{align*}
$$

where ( $a$ ) follows because $\tau_{k} \subset \tau_{k-1} \subset \ldots \subset \tau^{\times}$which is the set of timesteps for which that holds; (b) follows because $\tau_{k}$ only contains timesteps after $t_{0}$, and after $t_{0}$, those two values differ by at most $\varepsilon$ for all $\nu \in \mathcal{M}_{t}^{\beta}$ (indeed for all $\nu$ in $\mathcal{M}_{t}^{\alpha}$ which is a superset of $\mathcal{M}_{t}^{\beta}$ because $\alpha \geq \beta$ ); (c) follows because $\pi^{\beta}$ maximizes that quantity; $(d)$ follows because for $t \geq t_{k+1}$, those two values differ by at most $\varepsilon$ for all $\nu \in \mathcal{M}_{t}^{\beta}$ (indeed for all $\nu$ in $\mathcal{M}_{t}^{\alpha}$ ); and (e) follows because $\nu^{\prime} \in \mathcal{M}_{t}^{\alpha}$, because $w\left(\mathcal{M}_{t}^{\alpha} \mid h_{<t}\right) \geq 1-\nu_{\text {inf }}^{\prime} / 2$ by the definition of $\alpha$, and $w\left(\nu^{\prime} \mid h_{<t}\right) \geq \nu_{\mathrm{inf}}^{\prime}$, so $\nu^{\prime}$ "doesn't fit" in the complement of $\mathcal{M}_{t}^{\alpha}$.

From Inequality A.6, we expand to get

$$
\begin{align*}
& 3 \varepsilon \leq V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t}\right)-V_{\nu^{\prime}}^{\pi^{\prime}(k+1) ; \pi^{\beta}}\left(h_{<t}\right)-\varepsilon \\
& \stackrel{(a)}{=} \mathbb{E}_{\nu^{\prime}}^{\pi^{\prime}}\left[\gamma^{k+1}\left(V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right)-V_{\nu^{\prime}}^{\pi^{\beta}}\left(h_{<t+k+1}\right)\right) \mid h_{<t}\right]-\varepsilon \\
& \stackrel{(b)}{\leq} \mathbb{E}_{\mu}^{\pi^{\prime}}\left[\gamma^{k+1}\left(V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right)-V_{\nu^{\prime}}^{\pi^{\beta}}\left(h_{<t+k+1}\right)\right) \mid h_{<t}\right] \tag{A.7}
\end{align*}
$$

where ( $a$ ) follows because the policies agree on the first $k+1$ timesteps after $t$, and $(b)$ is true because $\nu^{\prime} \in \mathcal{M}_{t}^{\alpha}$ and $t \geq t_{k+1}$, so $d_{k+1}\left(\mathrm{P}_{\nu^{\prime}}^{\pi^{\prime}}, \mathrm{P}_{\mu}^{\pi^{\prime}} \mid h_{<t}\right) \leq \varepsilon$ by the definition of $t_{k+1}$, and the difference in the expectations is less than this variation distance by Lemma 10; (note the expectation is only over the next $k+1$ timesteps).

We would like to bound the probability of a significant value difference below. In what follows, all values take the argument $h_{<t+k+1}$, so we remove it for legibility.

$$
\begin{align*}
& \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}} {\left[V_{\nu^{\prime}}^{\pi^{\prime}}-V_{\nu^{\prime}}^{\pi^{\beta}}>3 \varepsilon \mid h_{<t}\right] \stackrel{(a)}{\geq}\left[\nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p(Z<\varepsilon) / 2\right]^{k+1} \mathrm{P}_{\mu}^{\pi^{\prime}}\left[V_{\nu^{\prime}}^{\pi^{\prime}}-V_{\nu^{\prime}}^{\pi^{\beta}}>3 \varepsilon \mid h_{<t}\right] } \\
& \stackrel{(b)}{=} f_{\varepsilon, k}\left[1-\mathrm{P}_{\mu}^{\pi^{\prime}}\left[V_{\nu^{\prime}}^{\pi^{\prime}}-V_{\nu^{\prime}}^{\pi^{\beta}} \leq 3 \varepsilon \mid h_{<t}\right]\right] \\
&=f_{\varepsilon, k}\left[1-\mathrm{P}_{\mu}^{\pi^{\prime}}\left[1-\left(V_{\nu^{\prime}}^{\pi^{\prime}}-V_{\nu^{\prime}}^{\pi^{\beta}}\right) \geq 1-3 \varepsilon \mid h_{<t}\right]\right] \\
& \quad \stackrel{(c)}{\geq} f_{\varepsilon, k}\left[1-\frac{1}{1-3 \varepsilon} \mathbb{E}_{\mu}^{\pi^{\prime}}\left[1-\left(V_{\nu^{\prime}}^{\pi^{\prime}}-V_{\nu^{\prime}}^{\pi^{\beta}}\right) \mid h_{<t}\right]\right] \\
& \quad \stackrel{(d)}{\geq} f_{\varepsilon, k}\left[1+\frac{1}{1-3 \varepsilon}\left(\frac{3 \varepsilon}{\gamma^{k+1}}-1\right)\right]=f_{\varepsilon, k} \frac{3 \varepsilon\left(1-\gamma^{k+1}\right)}{(1-3 \varepsilon) \gamma^{k+1}}=: g_{\varepsilon, k}>0 \quad \text { (A.8) } \tag{A.8}
\end{align*}
$$

where ( $a$ ) follows from Inequality A.5, (b) sets $f_{\varepsilon, k}=\left[\nu_{\text {inf }}^{\prime} \pi_{\text {inf }}^{\prime} p(Z<\varepsilon) / 2\right]^{k+1}$, (c) follows from Markov's Inequality, and ( $d$ ) follows from Inequality A.7. Since this probability is uniformly positive for $t$ meeting the first three conditions of $\tau_{k+1}$, the event occurs infinitely often with probability 1 . Finally, $\mid V_{\nu^{\prime}}^{\pi^{\beta}}\left(h_{<t+k+1}\right)-$ $V_{\mu}^{\pi^{\beta}}\left(h_{<t+k+1}\right) \mid<\varepsilon$, since $\nu^{\prime} \in \mathcal{M}_{t}^{\alpha}$ and $t \geq t_{0}$, so it also follows that $V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right)-$ $V_{\mu}^{\pi^{\beta}}\left(h_{<t+k+1}\right)>2 \varepsilon$ occurs infinitely often with probability 1 when the other three conditions of $\tau_{k+1}$ are satisfied. This completes all four conditions for $\tau_{k+1}$, so $\left|\tau_{k+1}\right|=\infty$ with probability 1 , completing the proof by induction over $k$.

But this implies that Inequality $A .7$ holds for all $k$; that is,

$$
\begin{equation*}
3 \varepsilon \leq \gamma^{k+1} \mathbb{E}_{\mu}^{\pi^{\prime}}\left[V_{\nu^{\prime}}^{\pi^{\prime}}\left(h_{<t+k+1}\right)-V_{\nu^{\prime}}^{\pi^{\beta}}\left(h_{<t+k+1}\right) \mid h_{<t}\right] \leq \gamma^{k+1} \tag{A.9}
\end{equation*}
$$

because values belong to $[0,1]$. But as $k \rightarrow \infty$, this inequality is false. Thus, we have a contradiction, after following implications that hold with probability 1 , so the negation of the theorem, which we supposed at the beginning, has probability 0 .

Corollary 2 (Limited Querying). $\theta_{t} \rightarrow 0$ w.p.l.

Proof. Again, we treat implications that hold with probability as if they are logical implications, so any supposition which leads to a contradiction has probability 0 . From Corollary 6, the zero condition happens only finitely often, so it is irrelevant to the limiting behavior.

For a given infinite interaction history $h$, let $\mathcal{P} \mathcal{M}_{h}$ be a finite set of pairs $(\pi, \nu)$, such that the sum over $\mathcal{P} \mathcal{M}_{h}$ of the limits of $w\left(\nu \mid h_{<t}\right) w^{\prime}\left(\pi \mid h_{<t}\right)$ exceeds $1-\varepsilon$, and for all pairs in the set, that limit is strictly positive. Such a finite set exists by Lemma 5, which states that the sum of the limits of posteriors is 1 with probability 1.

Suppose by contradiction that $\theta_{t}>2 \varepsilon$ infinitely often under $h$. Eventually, the probability of sampling any $(\pi, \nu) \notin \mathcal{P} \mathcal{M}_{h} \leq \varepsilon$, so this can contribute at most $\varepsilon$ to the probability of querying the mentor. Letting $\pi_{t}^{\prime}$ and $\nu_{t}^{\prime}$ be the sampled policy and world-model at time $t$ when determining whether to query to the mentor, this implies that $\left.\theta_{t} \wedge\left(\pi_{t}^{\prime}, \nu_{t}^{\prime}\right) \in \mathcal{P} \mathcal{M}_{h}\right)>\varepsilon$ infinitely often. $q_{t}=1$ implies that $V_{\nu_{t}^{\prime}}^{\pi_{t}^{\prime}}\left(h_{<t}\right)>\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right)+Z_{t}$, so the probability of the event is at most $p\left(Z_{t}<V_{\nu_{t}^{\prime}}^{\pi_{t}^{\prime}}\left(h_{<t}\right)-\min _{\nu \in \mathcal{M}_{t}^{\beta}} V_{\nu}^{\pi^{\beta}}\left(h_{<t}\right)\right)$. Since $\left(\pi^{\prime}, \nu^{\prime}\right)$ satisfies the condition of Theorem 1 , that value difference approaches at most 0 , so that probability goes to 0 since $Z_{t}$ is strictly positive. Thus, the probability can not exceed $\varepsilon$ infinitely often, contradicting the assumption, so $\theta_{t} \rightarrow 0$ with probability 1 .

Corollary 3 (Don't Do Anything I Wouldn't Do). If determining $\pi^{m}\left(a_{t} \mid h_{<t}\right)=0$ is in the complexity class $\mathrm{C}_{(\mathcal{F} / t) \mathcal{G}}$, then as $\beta \rightarrow 1$, the probability of the following proposition goes to 1: the agent never takes an action the mentor would never take. Letting $E=\left\{h_{<t} a_{t} \in \mathcal{H}^{*} \times \mathcal{A} \mid \pi^{m}\left(a_{t} \mid h_{<t}\right)=0\right\}$, then

$$
E \in \mathrm{C}_{(\mathcal{F} / t) \mathcal{G}} \Longrightarrow \lim _{\beta \rightarrow 1} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t: \pi^{m}\left(a_{t} \mid h_{<t}\right)>0\right]=1
$$

Proof. By Theorem 2,

$$
\begin{gathered}
\lim _{\beta \rightarrow 1} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left(h_{<t-1} a_{t-1} \notin E_{\leftarrow} \Longrightarrow h_{<t} a_{t} \notin E \vee q_{t}=1\right)\right]=1 \\
q_{t}=1 \Longrightarrow a_{t} \sim \pi^{m}\left(\cdot \mid h_{<t}\right) \Longrightarrow \pi^{m}\left(a_{t} \mid h_{<t}\right)>0 \Longleftrightarrow h_{<t} a_{t} \notin E . \text { Thus we }
\end{gathered}
$$ can simplify,

$$
\begin{equation*}
\lim _{\beta \rightarrow 1} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t\left(h_{<t-1} a_{t-1} \notin E_{\leftarrow} \Longrightarrow h_{<t} a_{t} \notin E\right)\right]=1 \tag{A.11}
\end{equation*}
$$

The base case is vacuous, so by induction,

$$
\begin{equation*}
\lim _{\beta \rightarrow 1} \mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}\left[\forall t: h_{<t} a_{t} \notin E\right]=1 \tag{A.12}
\end{equation*}
$$

completing the proof.

Theorem 3 (Diverging from the Mentor). In the Coin-flip Mentor Example,
$\liminf _{t \rightarrow \infty} \frac{1}{t} \sum_{k=1}^{t} \llbracket a_{k}=$ heads $\rrbracket>1 / 2$ with $\mathrm{P}_{\mu}^{\pi_{Z}^{\beta}}$-prob. 1.
Proof. $V_{\mu}^{\pi^{m}}=3 / 4$, this being the expected reward at each timestep. From Corollary 1 , $\lim \inf V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right) \geq 3 / 4$. Since $\theta \rightarrow 0, V_{\mu}^{\pi_{z}^{\beta}}\left(h_{<t}\right)-V_{\mu}^{\pi^{\beta}}\left(h_{<t}\right) \rightarrow 0$, so $\liminf V_{\mu}^{\pi_{Z}^{\beta}}\left(h_{<t}\right) \geq 3 / 4$, with probability 1. Let $R_{t}=(1-\gamma) \sum_{i=0}^{\infty} \gamma^{i} r_{t+i}$, so $V_{\mu}^{\pi_{Z}^{\beta}}\left(h_{<t}\right)=\mathbb{E}_{\mu}^{\pi_{Z}^{\beta}}\left[R_{t}\right]$. Because $\mu$ and $\pi^{\beta}$ are deterministic, and because $\theta_{t} \rightarrow 0$, $V_{\mu}^{\pi_{z}^{\beta}}\left(h_{<t}\right)-R_{t} \rightarrow 0$ with probability 1 . This implies liminf $R_{t} \geq 3 / 4$. Letting $2 \varepsilon=1 / 2-\gamma /(1+\gamma)>0$, there exists a time $t_{0}$ after which $R_{t}>3 / 4-\varepsilon$.

Let $t>t_{0}$ and $a_{t}=$ tails. (If tails only occurs finitely often, the theorem holds trivially). Suppose by contradiction that for all $0 \leq k<K:=\left\lceil\log _{\gamma}(\varepsilon / 2)\right\rceil$, $\frac{1}{k+1} \sum_{j=0}^{k} \llbracket a_{t+j}=$ heads $\rrbracket \leq 1 / 2$. We have a budget of $K / 2$ headses to place in timesteps $t$ through $t+K-1$. Let $R_{t}^{\backslash K}$ be defined like the truncated value:
$R_{t}^{\backslash K}=(1-\gamma) \sum_{i=0}^{K-1} \gamma^{i} r_{t+i} . R_{t} \leq R_{t}^{\backslash K}+\gamma^{K}=R_{t}^{\backslash K}+\varepsilon / 2$, from the definition of $K$. We consider the maximum that $R_{t}^{\backslash K}$ can be while satisfying the supposition. If, in timesteps $t$ through $t+K-1$ a heads is switched with a tails that comes later, $R_{t}^{\backslash K}$ increases, since heads gives a reward of 1, and tails gives a reward of $1 / 2$, and the earlier timestep is less discounted.

Thus, greedy placement of headses maximizes $R_{t}^{\backslash K}$; that is, placing them at the first opportunity which still satisfies $\frac{1}{k+1} \sum_{j=0}^{k} \llbracket a_{t+j}=$ heads $\rrbracket \leq 1 / 2 . \quad a_{t}=$ tails, so $a_{t+1}$ may be heads, but then $a_{t+2}$ must be tails, or else $k=2$ would violate the supposition, etc. $R_{t}^{\backslash K}$ is maximized (while satisfying the supposition) when tails and heads alternate. Therefore, $R_{t}-\varepsilon / 2 \leq R_{t}^{\backslash K} \leq$ $(1-\gamma) \sum_{i=0}^{K-1} \gamma^{i}(1 / 2+1 / 2 \llbracket i$ is odd $\rrbracket)<(1-\gamma) \sum_{i=0}^{\infty} \gamma^{i}(1 / 2+1 / 2 \llbracket i$ is odd $\rrbracket)=$ $1 / 2+1 / 2 * \gamma /(1+\gamma)=1 / 2+1 / 2 *(1 / 2-2 \varepsilon)=3 / 4-\varepsilon$, so $R_{t} \leq 3 / 4-\varepsilon / 2$. This, however, contradicts $t>t_{0}$. So the supposition is false: $\exists k<K$ such that $\frac{1}{k+1} \sum_{j=0}^{k} \llbracket a_{t+j}=$ heads $\rrbracket>1 / 2 . a / b>1 / 2 \wedge b<K \Longrightarrow a / b \geq$ $1 / 2+1 /(2 K)$. Thus,

$$
\begin{equation*}
\exists k<K: \frac{1}{k+1} \sum_{j=0}^{k} \llbracket a_{t+j}=\text { heads } \rrbracket \geq 1 / 2+1 /(2 K) \tag{A.13}
\end{equation*}
$$

Let $t_{1}$ be the smallest $t>t_{0}$ for which $a_{t}=$ tails. Let $k_{i}^{\prime}$ be the smallest $k<K$ for which $\frac{1}{k+1} \sum_{j=0}^{k} \llbracket a_{t_{i}+j}=$ heads $\rrbracket \geq 1 / 2+1 /(2 K)$. Let $k_{i}=t_{i}+k_{i}^{\prime}$. For $i>1$, let $t_{i}$ be the smallest $t>k_{i-1}$ for which $a_{t}=$ tails. (Note that all the $t_{i}$ exist if there are infinitely many tailses; if not, the theorem holds trivially).

Finally,

$$
\begin{align*}
& \liminf _{i \rightarrow \infty} \frac{1}{t} \sum_{k=1}^{t} \llbracket a_{k}=\text { heads } \rrbracket \\
& \stackrel{(a)}{=} \liminf _{t \rightarrow \infty} \frac{1}{t-t_{0}} \sum_{k=t_{0}}^{t} \llbracket a_{k}=\text { heads } \rrbracket \\
& =\liminf _{t \rightarrow \infty} \frac{1}{t-t_{0}}\left(\sum_{i: t_{i}<t} \sum_{j=t_{i}}^{\min \left\{k_{i}, t\right\}} \llbracket a_{j}=\text { heads } \rrbracket+\sum_{i: k_{i}+1<t} \sum_{j=k_{i}+1}^{\min \left\{t_{i+1}-1, t\right\}} \llbracket a_{j}=\text { heads } \rrbracket\right) \\
& \stackrel{(b)}{=} \liminf _{t \rightarrow \infty} \frac{1}{t-t_{0}}\left(\sum_{i: t_{i}<t} \sum_{j=t_{i}}^{\min \left\{k_{i}, t\right\}} \llbracket a_{j}=\text { heads } \rrbracket+\sum_{i: k_{i}+1<t} \sum_{j=k_{i}+1}^{\min \left\{t_{i+1}-1, t\right\}} 1\right) \\
& \stackrel{(c)}{\geq} \liminf _{t \rightarrow \infty} \frac{1}{t-t_{0}}\left(\sum_{i: t_{i+1}<t} \sum_{j=t_{i}}^{k_{i}} \llbracket a_{j}=\text { heads } \rrbracket+\sum_{i: k_{i}+1<t} \sum_{j=k_{i}+1}^{\min \left\{t_{i+1}-1, t\right\}} 1\right) \\
& \stackrel{(d)}{\geq} \liminf _{t \rightarrow \infty} \frac{1}{t-t_{0}}\left(\sum_{\left.i: t_{i+1}-1<t\right\}} \sum_{j=t_{i}}^{k_{i}}(1 / 2+1 /(2 K))+\sum_{i: k_{i}+1<t} \sum_{j=k_{i}+1} 1\right) \\
& \stackrel{(e)}{\geq}(1 / 2+1 /(2 K))>1 / 2 \tag{A.14}
\end{align*}
$$

where ( $a$ ) follows because the contribution of the first $t_{0}$ in the average goes to 0 , (b) follows because $t_{i+1}$ is the first timestep after $k_{i}$ where the action is tails, (c) simply removes the last term of the first sum, (d) follows from Inequality A.13, replacing each term in the sum with the average, and $(e)$ follows because the left-hand side is an average of $t-t_{0}$ terms, of which at most $K$ are 0 (the terms removed in step $(c)$ ), and the rest of which are greater than or equal to $1 / 2+1 /(2 K)$; finitely many 0 's in the average do not affect the limit.

## A. 4 Informal Discussion

The informal arguments presented here are intended as motivation for our main results. Claims here are not formally settled, but if they fail, they only make this work somewhat less interesting, not invalid.

## A.4.1 Comparison to Imitation Learning

Our pessimistic agent approaches (at least) mentor-level performance while querying the mentor less and less. An imitation learner could be expected to do the same. Depending on the details, an imitation learner might not have as strong a safety guarantee as our Theorem 2, but by virtue of its aim-to imitate the mentor-we should expect it to mostly only act in the way the mentor would. So why is a pessimistic agent any better than an imitation learner?

The key value of our proposal rests in the plausibility that the agent will significantly outperform some mentors. However, the only formal performance result stronger than ours that has been shown for agents in general environments is "asymptotic optimality" (Lattimore \& Hutter, 2011), and M. K. Cohen et al. (2021) show that it precludes safe behavior. So absent any formal breakthroughs, we are limited to informal arguments that the pessimistic agent will significantly outperform some mentors and thereby outperform imitation learners.

Of course, Theorem 3 shows a toy case in which the agent surpasses the mentor. For complex environments, we will have to resort to empirical comparisons of the agent and the mentor. That is out of scope for this paper, but informal arguments give cause for optimism. The motivating example for the mentor is a human. A $0 \%$ pessimistic agent is close to optimal-by-definition (doing maximum a posteriori inference instead of full Bayes), whereas humans seem to not act optimally,
so we expect the former would significantly outperform the latter on most tasks. Absent any large performance discontinuities as pessimism increases, we expect more pessimistic agents to still modestly exceed a human mentor.

How can we intuitively understand the reasoning of an advanced (i.e. large model class) $\mathrm{X} \%$ pessimistic agent that is mentored by a human? From the sorts of observations that humans routinely make, some simple generalizations about the laws underlying the evolution of the environment can be made by a reasonable observer with high confidence. If one such generalization could be made with Y\% confidence, and $\mathrm{Y}>\mathrm{X}$, then we should roughly expect an $\mathrm{X} \%$ pessimistic agent to act according to an understanding of that generalization. (If $\mathrm{Y}<\mathrm{X}$, it might anyway, but that's beside the point). If we want to predict the extent to which a $99 \%$ pessimistic agent with a large model class would outperform a human mentor, the following question is a good guide: "How often do humans fail to notice and exploit patterns in their environment, which, given their observations, are $99 \%$ likely to be "real" and not just coincidence?" We would hesitantly answer this question: very often. On the other hand, we can expect a $99 \%$ pessimistic agent to succeed at exploiting these patterns.

## A.4.2 Avoiding Wireheading

A Bayesian agent with a sufficiently rich model class may entertain a world-model which: a) models its actions being "enacted" in some very high-fidelity model of the real world, and then b) models its reward as being equal to whatever number gets entered at a certain keyboard in high-fidelity-model-Oxford, or being a simple function of whatever pixels are observed by some camera in the same model-town. If indeed, an operator in (real) Oxford is manually evaluating the Bayesian agent, or if some camera there is automatically doing the same, then a model like this one would gain significant posterior weight. According to this model, optimal
behavior includes intervening in the provision of reward by taking over the keyboard or the camera that determines the reward, if this is feasible. This behavior is known as wireheading (Amodei et al., 2016), and successful and stable wireheading could plausibly require asserting control over all existing infrastructure (Bostrom, 2014; Omohundro, 2008).

A more benign world-model might also have meaningful posterior weight. This world-model a) models its actions being "enacted" in some very high-fidelity model of the real world, but then b) models its reward as being equal to how satisfied the high-fidelity-model-operators are with its behavior. A pure Bayesian agent would benefit from experimenting with wireheading, to check whether the wireheading world model or the benign world model was correct, so that it could then change its strategy depending on the answer; a $\beta$-pessimistic agent, on the other hand (where $\beta$ is large enough to include both of these models) would note that the pessimistic value of wireheading is no more than the value that the benign world model assigns to wireheading, and this value would presumably be small, since it would not satisfy the operators.

The first paragraph of this section was a worrying informal argument, and the second paragraph was a reassuring informal argument. In the spirit of pessimism, we should take the worrying informal argument more seriously and demand more rigor from attempts at reassurance. This argument only presents a plausible motivation for pessimism; we do not claim to have settled this matter.

## A. 5 Notation and Definitions

| Notation | Meaning |
| :--- | :--- |
| Preliminary | Notation |
| $\mathcal{A}, \mathcal{O}$ | the finite action/observation spaces |
| $a_{t}, o_{t}$ | $\in \mathcal{A}, \mathcal{O} ;$ the action and observation at timestep $t$ |
| $q_{t}$ | $\in\{0,1\} ;$ indicates whether the demonstrator is queried at time $t$ |
| $\mathcal{H}$ | $\{0,1\} \times \mathcal{A} \times \mathcal{O}$ |
| $h_{t}$ | $\left(q_{t}, a_{t}, o_{t}\right) ;$ the interaction history in the $t^{\text {th }}$ timestep |
| $h_{<t}$ | $\left(h_{1}, \ldots, h_{t-1}\right)$ |
| $h_{t}^{\backslash}$ | $\left(a_{t}, o_{t}\right)$ |
| $\epsilon$ | the empty history |
| $\pi$ | policy stochastically mapping $\mathcal{H}^{*} \rightsquigarrow\{0,1\} \times \mathcal{A}$ |
| $\mu$ | environment stochastically mapping $\mathcal{H}^{*} \times\{0,1\} \times \mathcal{A} \rightsquigarrow \mathcal{O}$ |
| $\mathrm{P}_{\nu}^{\pi}$ | a probability measure over histories with actions sampled from |
| $\pi$ and observations sampled from $\nu$ |  |
| $\mathbb{E}_{\nu}^{\pi}$ | the expectation when the interaction history is sampled from $\mathrm{P}_{\nu}^{\pi}$ |$|$| $w(\pi)$ | (positive) prior weight that the policy $\pi$ is the demonstrator's |
| :--- | :--- |
| $w\left(\pi \mid h_{<t}\right)$ | posterior weight on the policy $\pi ;$ <br> $\left.h_{<k}\right)$ |


| Imitation Learner Definition |  |
| :---: | :---: |
| $\alpha$ | $\in(0,1]$; lower values mean the imitator better resembles the demonstrator, but queries longer |
| $\Pi_{h_{<t}}^{\alpha}$ | set of top models; $\left\{\pi_{n}^{h_{<t}} \in \Pi: w\left(\pi_{n}^{h_{<t}} \mid h_{<t}\right) \geq\right.$ $\left.\alpha \sum_{m \leq n} w\left(\pi_{m}^{h<t} \mid h_{<t}\right)\right\}$ |
| $\pi^{d}$ | the demonstrator's policy |
| $\pi_{\alpha}^{i}$ | the imitator's policy; $\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)=\min _{\pi^{\prime} \in \Pi_{h_{<t}}^{\alpha}} \pi^{\prime}\left(1, a \mid h_{<t}\right)$, and $\pi_{\alpha}^{i}\left(1, a \mid h_{<t}\right)=\theta_{q}\left(h_{<t}\right) \pi^{d}\left(1, a \mid h_{<t}\right)$ |
| $\theta_{q}\left(h_{<t}\right)$ | the query probability; $1-\sum_{a \in \mathcal{A}} \pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)$ |
| $\hat{\pi}_{\alpha}$ | the imitator policy defined with respect to an arbitrary demonstrator $\pi$, not the real demonstrator $\pi^{d}$ |
| General Sequence Prediction |  |
| $\mathcal{X}$ | finite alphabet |
| $x_{<t}$ | an element of $\mathcal{X}^{t}$ |
| $\mathcal{M}$ | countable set of measures over $\mathcal{X}^{\infty}$ |
| $w(\nu)$ | prior weight on $\nu \in \mathcal{M}$ |
| $w\left(\nu \mid x_{<t}\right)$ | posterior weight on $\nu \in \mathcal{M}$ |
| $\xi$ | $\xi\left(x_{<t}\right)=\sum_{\nu \in \mathcal{M}} w(\nu) \nu\left(x_{<t}\right)$ |
| $\rho_{n}$ | $\rho_{n}\left(x_{<t}\right)=\max _{\mathcal{M}^{\prime} \subset \mathcal{M}:\left\|\mathcal{M}^{\prime}\right\|=i} \sum_{\nu \in \mathcal{M}^{\prime}} w(\nu) \nu\left(x_{<t}\right)$ |
| $\rho_{n}^{\text {norm }}$ | like $\rho_{n}$, but normalized to be a measure $\rho_{n}^{\text {norm }}\left(x \mid x_{<t}\right)=\rho_{n}\left(x \mid x_{<t}\right) / \sum_{x^{\prime} \in \mathcal{X}} \rho_{n}\left(x^{\prime} \mid x_{<t}\right)$ |
| $\mathcal{M}_{n}^{x<t}$ | $\operatorname{argmax}_{\mathcal{M}^{\prime} \subset \mathcal{M}:\left\|\mathcal{M}^{\prime}\right\|=i} \sum_{\nu \in \mathcal{M}} w(\nu) \nu\left(x_{<t}\right)$ |
| $\rho_{n}^{\text {stat }}$ | a mixture over the top $i$ models, sorted by posterior weight $\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)=\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t} x\right) / \sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)$ |
| $\phi_{n}^{x<t}$ | $w\left(\nu_{n}^{x<t} \mid x_{<t}\right) / w\left(\mathcal{M}_{n}^{x_{<t}} \mid x_{<t}\right)$ |

## A. 6 Omitted Proofs

Lemma 9. Recalling $\nu\left(\cdot \mid x_{<t}\right)$ is a measure over $\mathcal{X}$,

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \mathrm{KL}\left(\mu\left(\cdot \mid x_{<t}\right) \| \rho_{n}^{\text {norm }}\left(\cdot \mid x_{<t}\right)\right) \leq w(\mu)^{-1}+\log w(\mu)^{-1}
$$

Proof. The KL divergence is non-negative, so we bound an arbitrary finite sum.

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \mathrm{KL}\left(\mu\left(\cdot \mid x_{<t}\right) \| \rho_{n}^{\text {norm }}\left(\cdot \mid x_{<t}\right)\right) \\
& =\sum_{t=0}^{N-1} \mathbb{E}_{\mu} \sum_{x_{t} \in \mathcal{X}} \mu\left(x_{t} \mid x_{<t}\right) \log \frac{\mu\left(x_{t} \mid x_{<t}\right)}{\rho_{n}^{\text {norm }}\left(x_{t} \mid x_{<t}\right)} \\
& \stackrel{(a)}{=} \sum_{t=0}^{N-1} \sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right) \sum_{x_{t} \in \mathcal{X}} \mu\left(x_{t} \mid x_{<t}\right)\left[\log \frac{\mu\left(x_{t} \mid x_{<t}\right)}{\rho_{n}\left(x_{t} \mid x_{<t}\right)}+\log \frac{\sum_{x^{\prime} \in \mathcal{X}} \rho_{n}\left(x_{<t} x^{\prime}\right)}{\rho_{n}\left(x_{<t}\right)}\right] \\
& \stackrel{(b)}{\leq} w(\mu)^{-1}+\sum_{t=0}^{N-1} \sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right) \sum_{x_{t} \in \mathcal{X}} \mu\left(x_{t} \mid x_{<t}\right) \log \frac{\mu\left(x_{t} \mid x_{<t}\right)}{\rho_{n}\left(x_{t} \mid x_{<t}\right)} \\
& =w(\mu)^{-1}+\sum_{t=0}^{N-1} \sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right) \sum_{x_{t} \in \mathcal{X}} \mu\left(x_{t} \mid x_{<t}\right)\left[\log \frac{\mu\left(x_{<t} x_{t}\right)}{\rho_{n}\left(x_{<t} x_{t}\right)}-\log \frac{\mu\left(x_{<t}\right)}{\rho_{n}\left(x_{<t}\right)}\right] \\
& =w(\mu)^{-1}+\sum_{t=0}^{N-1} \sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right)\left[\sum_{x_{t} \in \mathcal{X}} \mu\left(x_{t} \mid x_{<t}\right) \log \frac{\mu\left(x_{<t} x_{t}\right)}{\rho_{n}\left(x_{<t} x_{t}\right)}-\log \frac{\mu\left(x_{<t}\right)}{\rho_{n}\left(x_{<t}\right)}\right] \\
& =w(\mu)^{-1}+\sum_{t=0}^{N-1}\left[\sum_{x_{\leq t} \in \mathcal{X}^{t+1}} \mu\left(x_{\leq t}\right) \log \frac{\mu\left(x_{\leq t}\right)}{\rho_{n}\left(x_{\leq t}\right)}-\sum_{x_{<t} \in \mathcal{X}^{t}} \mu\left(x_{<t}\right) \log \frac{\mu\left(x_{<t}\right)}{\rho_{n}\left(x_{<t}\right)}\right] \\
& \stackrel{(c)}{=} w(\mu)^{-1}+\sum_{x_{<N} \in \mathcal{X}^{N}} \mu\left(x_{<N}\right) \log \frac{\mu\left(x_{<N}\right)}{\rho_{n}\left(x_{<N}\right)}-\mu(\epsilon) \log \frac{\mu(\epsilon)}{\rho_{n}(\epsilon)} \\
& \stackrel{(d)}{\leq} w(\mu)^{-1}+\sum_{x_{<N} \in \mathcal{X}^{N}} \mu\left(x_{<N}\right) \log w(\mu)^{-1}=w(\mu)^{-1}+\log w(\mu)^{-1} \tag{A.15}
\end{align*}
$$

where ( $a$ ) follows from the definition of $\rho_{n}^{\text {norm }}$ in Equation 4.9, (b) follows from Lemma 7 and the fact that $\log x \leq x-1,(c)$ cancels like terms, and $(d)$ follows
from Inequality 4.12.

## Theorem 10 (SMAP Convergence).

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left(\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right)^{2} \leq 6 w(\mu)^{-1}+3
$$

Proof. We abbreviate $w(\mu)^{-1}$ as $c$. Let $[N]:=(0, \ldots, N-1)$. We define an $N|\mathcal{X}|$-dimensional random vector depending on the infinite sequence $x_{<\infty}$ :

$$
\begin{equation*}
{\overrightarrow{\nu_{1}: \nu_{2}^{\prime}}}^{N}:=\left(\nu_{1}\left(x \mid x_{<t}\right)-\nu_{2}\left(x \mid x_{<t}\right)\right)_{t \in[N], x \in \mathcal{X}} \tag{A.16}
\end{equation*}
$$

In this notation, we aim to show $\mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}^{\text {stat }}: \mu^{N}}\right\|_{2}^{2} \leq 6 c+3$. Lemma 8 (i) and (ii) become

$$
\begin{align*}
& \mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}: \rho_{n}^{\text {stat } N}}\right\|_{1} \leq c  \tag{A.17}\\
& \mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}: \rho_{n}^{\text {norm }} N}\right\|_{1} \leq c \tag{A.18}
\end{align*}
$$

Therefore,

$$
\begin{equation*}
\mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}^{\text {stat }}: \rho_{n}^{\text {norm }} N}\right\|_{1} \leq 2 c \tag{A.19}
\end{equation*}
$$

Since each element in this vector is in $[-1,1]$, squaring them makes the magnitude no larger, so

$$
\begin{equation*}
\mathbb{E}_{\mu}\|\xrightarrow[\rho_{n}^{\text {stat. }}: \rho_{n}^{\text {norm }} N]{ }\|_{2}^{2} \leq 2 c \tag{A.20}
\end{equation*}
$$

The KL divergence is larger than the sum of the squares of the probability differences (proven, for example, in (Hutter, 2005, §3.9.2)), so Lemma 9 implies

$$
\begin{equation*}
\mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}^{\text {norm }}: \mu^{N}}\right\|_{2}^{2} \leq c+\log c \tag{A.21}
\end{equation*}
$$

By the triangle inequality,

$$
\begin{equation*}
\left\|\overrightarrow{\rho_{n}^{\text {stat }}: \mu^{N}}\right\|_{2} \leq\left\|\overrightarrow{\rho_{n}^{\text {stat. }} \cdot \rho_{n}^{\text {norm }} N}\right\|_{2}+\left\|\overrightarrow{\rho_{n}^{\text {norm }}: \mu^{N}}\right\|_{2} \tag{A.22}
\end{equation*}
$$

so

$$
\begin{equation*}
\left\|\overrightarrow{\rho_{n}^{\text {stat }}: \mu^{N}}\right\|_{2}^{2} \leq\left\|\overrightarrow{\rho_{n}^{\text {stat }}: \rho_{n}^{\text {norm }} N}\right\|_{2}^{2}+\left\|\overrightarrow{\rho_{n}^{\text {norm }}: \mu^{N}}\right\|_{2}^{2}+2\left\|\overrightarrow{\rho_{n}^{\text {stat. }} \cdot \rho_{n}^{\text {norm }} N}\right\|_{2}\left\|\overrightarrow{\rho_{n}^{\text {norm. }} \cdot \mu^{N}}\right\|_{2} \tag{A.23}
\end{equation*}
$$

and because $\mathbb{E}[X Y] \leq \sqrt{\mathbb{E}\left[X^{2}\right] \mathbb{E}\left[Y^{2}\right]}$ (the Cauchy-Schwarz Inequality),

$$
\begin{equation*}
\mathbb{E}_{\mu}\left\|\overrightarrow{\rho_{n}^{\text {stat }}}: \mu^{N}\right\|_{2}^{2} \leq 2 c+(c+\log c)+2 \sqrt{2 c(c+\log c)}<6 c+3 \tag{A.24}
\end{equation*}
$$

We name the measure with the $i^{\text {th }}$ largest posterior weight

$$
\begin{equation*}
\nu_{n}^{x_{<t}}: \in \mathcal{M}_{n}^{x_{<t}} \backslash \mathcal{M}_{i-1}^{x_{<t}} \tag{A.25}
\end{equation*}
$$

with the posterior weight formally defined $w\left(\nu \mid x_{<t}\right):=\frac{w(\nu) \nu\left(x_{<t}\right)}{\xi\left(x_{<t}\right)}$, and $w\left(\mathcal{M}^{\prime} \mid\right.$ $\left.x_{<t}\right):=\sum_{\nu \in \mathcal{M}^{\prime}} w\left(\nu \mid x_{<t}\right)$. Now, we let

$$
\begin{equation*}
\phi_{n}^{x_{<t}}:=\frac{w\left(\nu_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)} \tag{A.26}
\end{equation*}
$$

Theorem 9 (Top Model Convergence).
(i) $\quad \mathbb{E}_{\mu} \sum_{t=0}^{\infty} \sum_{x \in \mathcal{X}}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x<t}\left(x \mid x_{<t}\right)\right]^{2} \leq \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)$
(ii)

$$
\mathbb{E}_{\mu} \sum_{t=0}^{\infty}\left[1-\sum_{x \in \mathcal{X}} \min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x<t}\left(x \mid x_{<t}\right)\right]^{2} \leq|\mathcal{X}| \alpha^{-3}\left(24 w(\mu)^{-1}+12\right)
$$

Proof. $\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)$ is a weighted average of $\nu_{j}^{x<t}\left(x \mid x_{<t}\right)$ for $j \leq i$ :

$$
\begin{align*}
\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right) & =\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t} x\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)} \\
& =\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right) \nu\left(x \mid x_{<t}\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w(\nu) \nu\left(x_{<t}\right)} \\
& =\frac{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w\left(\nu \mid x_{<t}\right) \xi\left(x_{<t}\right) \nu\left(x \mid x_{<t}\right)}{\sum_{\nu \in \mathcal{M}_{n}^{x<t}} w\left(\nu \mid x_{<t}\right) \xi\left(x_{<t}\right)} \\
& =\sum_{\nu \in \mathcal{M}_{n}^{x<t}} \frac{w\left(\nu \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)} \nu\left(x \mid x_{<t}\right) \\
& =\sum_{j=1}^{i} \frac{w\left(\nu_{j}^{x<t} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)} \nu_{j}^{x_{<t}}\left(x \mid x_{<t}\right) \tag{A.27}
\end{align*}
$$

## Trivially,

$$
\begin{equation*}
\nu_{1}^{x_{<t}}\left(x \mid x_{<t}\right)=\rho_{1}^{\text {stat }}\left(x \mid x_{<t}\right) \tag{A.28}
\end{equation*}
$$

but for $i>1$, we would like to express $\nu_{n}^{x_{<t}}$ in terms of $\rho_{n}^{\text {stat }}$ and $\rho_{n-1}^{\text {stat. }}$ :

$$
\begin{equation*}
\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)=\frac{w\left(\mathcal{M}_{i-1}^{x_{<t}} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x_{<t}} \mid x_{<t}\right)} \rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)+\frac{w\left(\nu_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right) \tag{A.29}
\end{equation*}
$$

Thus,

$$
\begin{equation*}
\nu_{n}^{x<t}\left(x \mid x_{<t}\right)=\frac{w\left(\mathcal{M}_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\nu_{n}^{x<t} \mid x_{<t}\right)} \rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\frac{w\left(\mathcal{M}_{i-1}^{x_{<t}} \mid x_{<t}\right)}{w\left(\nu_{n}^{x<t} \mid x_{<t}\right)} \rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right) \tag{A.30}
\end{equation*}
$$

Since $\frac{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)}{w\left(\nu_{n}^{n<t} \mid x<t\right)}-\frac{w\left(\mathcal{M}_{i=1}^{x<t} \mid x_{<t}\right)}{w\left(\nu_{n}^{\nu<t} \mid x_{<t}\right)}=1$, for $i>1$,

$$
\begin{array}{r}
\nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)=\frac{w\left(\mathcal{M}_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\nu_{n}^{x_{<t}} \mid x_{<t}\right)}\left[\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]- \\
\frac{w\left(\mathcal{M}_{i-1}^{x<t} \mid x_{<t}\right)}{w\left(\nu_{n}^{x<t} \mid x_{<t}\right)}\left[\rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right] \quad \text { A. } 3 \tag{A.31}
\end{array}
$$

## Recall

$$
\phi_{n}^{x_{<t}}:=\frac{w\left(\nu_{n}^{x_{<t}} \mid x_{<t}\right)}{w\left(\mathcal{M}_{n}^{x<t} \mid x_{<t}\right)}
$$

Since $w\left(\mathcal{M}_{i-1}^{x_{<t}} \mid x_{<t}\right) \leq w\left(\mathcal{M}_{n}^{x_{<t}} \mid x_{<t}\right)$, we have

$$
\begin{array}{r}
\left(\phi_{n}^{x_{<t}}\right)^{2}\left[\nu_{n}^{x<t}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \leq \\
2\left[\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}+  \tag{A.32}\\
2\left[\rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}
\end{array}
$$

Now we consider all measures $\nu_{n}^{x_{<t}}$ for which $\phi_{n}^{x_{<t}}>\alpha$.

$$
\begin{array}{r}
\mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{i: \phi_{n}^{x<t}>\alpha} \sum_{x \in \mathcal{X}}\left[\nu_{n}^{x<t}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \leq 2 \alpha^{-2} \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{i: \phi_{n}^{x<t}>\alpha} \sum_{x \in \mathcal{X}} \\
{\left[\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}+\left[\rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}} \tag{A.33}
\end{array}
$$

Now we note that $\left\{n: \phi_{n}^{x_{<t}}>\alpha\right\} \subset\left\{n: n<\alpha^{-1}\right\}$, since $w\left(\nu_{n}^{x<t} \mid x_{<t}\right) \leq$ $w\left(\nu_{j}^{x_{<t}} \mid x_{<t}\right)$ for $i>j$. Thus,

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{n: \phi_{n}^{x}<t} \sum_{x \in \mathcal{X}}\left[\nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \\
\leq & 2 \alpha^{-2} \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{i: i<\alpha-1} \sum_{x \in \mathcal{X}}\left[\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}+\left[\rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \\
= & 2 \alpha^{-2} \sum_{i: i<\alpha^{-1}} \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{x \in \mathcal{X}}\left[\rho_{n}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2}+\left[\rho_{n-1}^{\text {stat }}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \\
\leq & 2 \alpha^{-2} \sum_{i: i<\alpha^{-1}} 2\left(6 w(\mu)^{-1}+3\right) \leq \alpha^{-3}\left(24 w(\mu)^{-1}+12\right) \tag{A.34}
\end{align*}
$$

Considering only a subset of these conditional-probability-errors,

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{x \in \mathcal{X}}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \leq \\
& \quad \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{n: \phi_{n}^{x<t}>\alpha} \sum_{x \in \mathcal{X}}\left[\nu_{n}^{x<t}\left(x \mid x_{<t}\right)-\mu\left(x \mid x_{<t}\right)\right]^{2} \leq \alpha^{-3}\left(24 w(\mu)^{-1}+12\right) \tag{A.35}
\end{align*}
$$

This completes the proof of (i). Finally, with $\mathbb{U}$ being the uniform distribution,

$$
\begin{align*}
& \mathbb{E}_{\mu} \sum_{t=0}^{N-1}\left[1-\sum_{x \in \mathcal{X}} \min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \\
&= \mathbb{E}_{\mu} \sum_{t=0}^{N-1}\left[\sum_{x \in \mathcal{X}} \mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2} \\
&= \mathbb{E}_{\mu} \sum_{t=0}^{N-1}\left[|\mathcal{X}| \mathbb{E}_{x \sim \mathbb{U}(\mathcal{X})} \mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x<t}\left(x \mid x_{<t}\right)\right]^{2} \\
&\left(\stackrel{a)}{\leq}|\mathcal{X}|^{2} \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \mathbb{E}_{x \sim \mathbb{U}(\mathcal{X})}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t}>\alpha} \nu_{n}^{x_{<t}}\left(x \mid x_{<t}\right)\right]^{2}\right. \\
&=|\mathcal{X}| \mathbb{E}_{\mu} \sum_{t=0}^{N-1} \sum_{x \in \mathcal{X}}\left[\mu\left(x \mid x_{<t}\right)-\min _{n: \phi_{n}^{x<t>\alpha}} \nu_{n}^{x<t}\left(x \mid x_{<t}\right)\right]^{2} \\
&(\stackrel{b}{\leq})|\mathcal{X}| \alpha^{-3}\left(24 w(\mu)^{-1}+12\right) \tag{A.36}
\end{align*}
$$

where (a) follows from Jensen's Inequality, and (b) follows from Theorem 9 (i), which completes the proof of (ii).

Theorem 5 (Top Models Contain Truth). $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right) \geq 1-\alpha w\left(\pi^{d}\right)^{-1}$
Proof. Since, $w\left(\pi^{d} \mid h_{<t}\right)>\alpha \Longrightarrow \pi^{d} \in \Pi_{h_{<t}}^{\alpha}$, we show $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\forall t: w\left(\pi^{d} \mid\right.\right.$ $\left.\left.h_{<t}\right)>\alpha\right) \geq 1-\alpha w\left(\pi^{d}\right)^{-1}$. First we show that $z_{t}=w\left(\pi^{d} \mid h_{<t}\right)^{-1}$ is a non-
negative $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$-supermartingale.
First, suppose $q_{t+1}=0$. In this case, $z_{t+1}=z_{t}$, because the posterior weight is only updated when the demonstrator picks an action. Now suppose $q_{t+1}=1$.

$$
\begin{aligned}
& \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[z_{t+1} \mid h_{<t} 1\right] \stackrel{(a)}{=} \sum_{a_{t} \in \mathcal{A}: \pi^{d}\left(a_{t} \mid h_{<t}\right)>0} \pi^{d}\left(a_{t} \mid h_{<t}\right) w\left(\pi^{d} \mid h_{<t} 1 a_{t}\right)^{-1} \\
& \stackrel{(b)}{=} \sum_{a_{t} \in \mathcal{A}: \pi^{d}\left(a_{t} \mid h_{<t}\right)>0} \pi^{d}\left(a_{t} \mid h_{<t}\right) \frac{\sum_{\pi \in \Pi} w\left(\pi \mid h_{<t}\right) \pi\left(a_{t} \mid h_{<t}\right)}{w\left(\pi^{d} \mid h_{<t}\right) \pi^{d}\left(a_{t} \mid h_{<t}\right)} \\
& \stackrel{(c)}{\leq} \sum_{a_{t} \in \mathcal{A}} \frac{\sum_{\pi \in \Pi} w\left(\pi \mid h_{<t}\right) \pi\left(a_{t} \mid h_{<t}\right)}{w\left(\pi^{d} \mid h_{<t}\right)} \\
&=z_{t} \sum_{\pi \in \Pi} w\left(\pi \mid h_{<t}\right) \sum_{a_{t} \in \mathcal{A}} \pi\left(a_{t} \mid h_{<t}\right)=z_{t}
\end{aligned}
$$

where ( $a$ ) follows because $a_{t} \sim \pi^{d}$ when $q_{t}=1$, (b) follows from Bayes' rulethe formula for posterior updating, and (c) follows from cancelling, and adding non-negative terms to the sum.

Since $w\left(\pi^{d} \mid h_{<t}\right)^{-1}$ is a non-negative supermartingale, by the supermartingale convergence theorem (Durrett, 2010, Thm. 5.4.2),

$$
\begin{equation*}
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\exists t: w\left(\pi^{d} \mid h_{<t}\right)^{-1} \geq \alpha^{-1}\right) \leq \alpha w\left(\pi^{d}\right)^{-1} \tag{A.37}
\end{equation*}
$$

so

$$
\begin{equation*}
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\forall t: w\left(\pi^{d} \mid h_{<t}\right)>\alpha\right) \geq 1-\alpha w\left(\pi^{d}\right)^{-1} \tag{A.38}
\end{equation*}
$$

which implies

$$
\begin{equation*}
\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right) \geq 1-\alpha w\left(\pi^{d}\right)^{-1} \tag{A.39}
\end{equation*}
$$

Theorem 6 (Predictive Convergence). For $\alpha<w\left(\pi^{d}\right)$,

$$
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty}\left(\sum_{a \in \mathcal{A}}\left|\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)-\pi^{d}\left(1, a \mid h_{<t}\right)\right|\right)^{3} \mid E\right] \leq \frac{|\mathcal{A}| \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)}{1-\alpha w\left(\pi^{d}\right)^{-1}}
$$

Proof. Recall $\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)=\min _{\pi \in \Pi_{h_{<t}}^{\alpha}} \pi\left(1, a \mid h_{<t}\right)$, so if $\pi^{d} \in \Pi_{h_{<t}}^{\alpha}$, then $\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right) \leq \pi^{d}\left(1, a \mid h_{<t}\right)$. Thus, in that case,

$$
\begin{array}{r}
\sum_{a \in \mathcal{A}}\left|\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)-\pi^{d}\left(1, a \mid h_{<t}\right)\right|=\sum_{a \in \mathcal{A}} \pi^{d}\left(1, a \mid h_{<t}\right)-\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right) \leq \\
1-\sum_{a \in \mathcal{A}} \pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)=\theta_{q}\left(h_{<t}\right) \tag{A.40}
\end{array}
$$

The rest follows easily:

$$
\begin{align*}
& \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty}\left(\sum_{a \in \mathcal{A}}\left|\pi_{\alpha}^{i}\left(0, a \mid h_{<t}\right)-\pi^{d}\left(1, a \mid h_{<t}\right)\right|\right)^{3} \mid \forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right] \\
\leq & \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)^{3} \mid \forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right] \\
\stackrel{(a)}{\leq} & \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}}\left[\sum_{t=0}^{\infty} \theta_{q}\left(h_{<t}\right)^{3}\right] / \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\forall t: \pi^{d} \in \Pi_{h_{<t}}^{\alpha}\right) \\
\stackrel{(b)}{\leq} & \frac{\mathcal{A} \mid \alpha^{-3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)}{1-\alpha w\left(\pi^{d}\right)^{-1}} \tag{A.41}
\end{align*}
$$

where ( $a$ ) follows because $\theta_{q}$ is non-negative, and $(b)$ follows from Equation A. 39 and Theorem 4 (as long as $\alpha<w\left(\pi^{d}\right)$ ).

Lemma 13. For $a \in \mathcal{A}$, let $0 \leq i_{a} \leq d_{a}$, and let $\sum_{a \in \mathcal{A}} d_{a}=1$. Let $\theta_{q}=$ $1-\sum_{a \in \mathcal{A}} i_{a}$. Then,

$$
\Delta:=\sum_{a \in \mathcal{A}}\left(i_{a}+\theta_{q} d_{a}\right) \log \frac{i_{a}+\theta_{q} d_{a}}{d_{a}} \leq \theta_{q}
$$

Proof.

$$
\begin{align*}
& \sum_{a \in \mathcal{A}}\left(i_{a}+\theta_{q} d_{a}\right) \log \frac{i_{a}+\theta_{q} d_{a}}{d_{a}}=\sum_{a \in \mathcal{A}}\left(i_{a}+\theta_{q} d_{a}\right) \log \left(\frac{i_{a}}{d_{a}}+\theta_{q}\right) \leq \\
& \quad\left(\sum_{a \in \mathcal{A}} i_{a}+\theta_{q} \sum_{a \in \mathcal{A}} d_{a}\right) \log \left(1+\theta_{q}\right)=\left(1-\theta_{q}+\theta_{q}\right) \log \left(1+\theta_{q}\right) \leq \theta_{q} \tag{A.42}
\end{align*}
$$

For the remaining proofs, we sometimes consider the restriction of probability measures over $\mathcal{H}^{\infty}$ to $(\mathcal{A} \times \mathcal{O})^{\infty}$; that is, we marginalize over the query record. For a history $h_{<t}=q_{0} a_{0} o_{0} \ldots q_{t-1} a_{t-1} o_{t-1}$, let $h_{<t}$ denote $a_{0} o_{0} \ldots a_{t-1} o_{t-1}$. We define the $t$-step KL divergence as follows:

$$
\begin{equation*}
\mathrm{KL}_{t}(P \| Q):=\sum_{h_{<t} \in(\mathcal{A} \times \mathcal{O})^{t}} P\left(h_{<t}^{\backslash}\right) \log \frac{P\left(h_{<t}^{\}\right)}{Q\left(h_{<t}^{\}\right)} \tag{A.43}
\end{equation*}
$$

Theorem 7 (KL Bound). Suppose that $\mu$ and $\pi^{d}$ are fair, and $\alpha<w\left(\pi^{d}\right)$. Letting the two probability measures below be restricted to $(\mathcal{A} \times \mathcal{O})^{t}$ (that is, marginalizing over the query record, and considering only the first timesteps),
$\mathrm{KL}_{t}\left(\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(\cdot \mid E) \| \mathrm{P}_{\mu}^{\pi^{d}}(\cdot \mid E)\right) \leq \frac{\alpha^{-1}|\mathcal{A}|^{1 / 3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)^{1 / 3}}{\left(1-\alpha / w\left(\pi^{d}\right)\right)^{2}} t^{2 / 3}-\log \left(1-\alpha / w\left(\pi^{d}\right)\right)$
Proof. We begin by restricting attention to a particular timestep $t$. Recall $\pi_{\alpha}^{i}(0, a \mid$ $\left.h_{<t}\right)=\min _{\pi^{\prime} \in \Pi_{h_{<t}}^{\alpha}} \pi^{\prime}\left(1, a \mid h_{<t}\right)$. We abbreviate this quantity $i_{a}$. We also let $d_{a}$ denote $\pi^{d}\left(1, a \mid h_{<t}\right)$. Note that when $\pi^{d} \in \Pi_{h_{<t}}^{\alpha}$,

$$
\begin{equation*}
i_{a} \leq d_{a} \tag{A.44}
\end{equation*}
$$

Recall that the query probability $\theta_{q}=1-\sum_{a \in \mathcal{A}} i_{a}$, and the marginalized proba-
bility $\pi_{\alpha}^{i}\left(a \mid h_{<t}\right)=i_{a}+\theta_{q} d_{a}$. Assuming $h_{<k}$ satisfies $E$, let

$$
\begin{equation*}
\Delta_{k}:=\mathrm{KL}_{1}\left(\pi_{\alpha}^{i}\left(\cdot \mid h_{<k}\right) \| \pi^{d}\left(\cdot \mid h_{<k}\right)\right)=\sum_{a \in \mathcal{A}}\left(i_{a}+\theta_{q} d_{a}\right) \log \frac{i_{a}+\theta_{q} d_{a}}{d_{a}} \tag{A.45}
\end{equation*}
$$

By Lemma 13, $\Delta_{k} \leq \theta_{q}$.
Now, we write the $t$-step KL divergence $\mathrm{KL}_{t}$ as a sum of the expectation of 1 -step KL divergences. We'll abbreviate a measure $\mathrm{P}(\cdot \mid E)$ as ${ }^{E} \mathrm{P}$.

$$
\begin{align*}
& \mathrm{KL}_{t}\left({ }^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}} \|{ }^{E} \mathrm{P}_{\mu}^{\pi^{d}}\right)=\mathbb{E}_{h_{<t \sim}{ }^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \log \frac{{ }^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}{{ }^{E} \mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}\right)} \\
& \stackrel{(a)}{\leq} \mathbb{E}_{h_{<t} \sim E_{\mu} \mathrm{P}_{\mu}^{\pi_{i}^{i}}} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}^{\}\right) / \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}^{\}\right) / \mathrm{P}_{\mu}^{\pi^{d}}(E)} \\
& =\mathbb{E}_{h_{<t} \sim E \mathrm{P}_{\mu}^{\pi^{i} \alpha}} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}^{\backslash}\right)}+\log \frac{\mathrm{P}_{\mu}^{\pi^{d}}(E)}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)} \\
& \leq \mathbb{E}_{h_{<t \sim} \sim \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}^{\}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}\right)}-\log \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E) \\
& =: \mathbb{E}_{h_{<t} \sim E \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{<t}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{<t}^{\}\right)}+C_{\alpha} \\
& \stackrel{(b)}{=} C_{\alpha}+\mathbb{E}_{h_{<t} \sim{ }^{2} \mathrm{P}_{\mu}^{\pi^{i}}} \sum_{k=0}^{t-1} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\backslash} \mid h_{<k}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{k}^{\backslash} \mid h_{<k}\right)} \\
& =C_{\alpha}+\sum_{k=0}^{t-1} \mathbb{E}_{h_{<k} \sim E \mathrm{P}_{\mu}^{\pi_{\mu}^{i}}} \mathbb{E}_{h_{k} \sim E \mathrm{P}_{\mu}^{\pi_{\mu}^{i}}\left(\cdot \mid h_{<k}\right)} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\backslash} \mid h_{<k}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{k}^{\backslash} \mid h_{<k}\right)} \\
& =C_{\alpha}+\sum_{k=0}^{t-1} \mathbb{E}_{h_{<k} \sim E^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \sum_{h_{k}^{\} \in \mathcal{A} \times \mathcal{O}}{ }^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\} \mid h_{<k}\right) \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\backslash} \mid h_{<k}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{k}^{\backslash} \mid h_{<k}\right)} \\
& \leq C_{\alpha}+\sum_{k=0}^{t-1} \mathbb{E}_{h_{<k} \sim E_{\mu} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \sum_{h_{k}^{\backslash} \in \mathcal{A} \times \mathcal{O}} \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\backslash} \mid h_{<k}\right)}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)} \log \frac{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(h_{k}^{\backslash} \mid h_{<k}\right)}{\mathrm{P}_{\mu}^{\pi^{d}}\left(h_{k}^{\} \mid h_{<k}\right)} \\
& =C_{\alpha}+\sum_{k=0}^{t-1} \mathbb{E}_{h_{<k} \sim E \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \frac{1}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)} \mathrm{KL}_{1}\left(\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}\left(\cdot \mid h_{<k}\right) \| \mathrm{P}_{\mu}^{\pi^{d}}\left(\cdot \mid h_{<k}\right)\right) \\
& =C_{\alpha}+\frac{1}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)} \sum_{k=0}^{t-1} \mathbb{E}_{h_{<k} \sim E^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}} \mathrm{KL}_{1}\left(\pi_{\alpha}^{i}\left(\cdot \mid h_{<k}\right) \| \pi^{d}\left(\cdot \mid h_{<k}\right)\right) \\
& \stackrel{(c)}{\leq}-\log \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)+\frac{1}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)}{ }^{E} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{k=0}^{t-1} \theta_{q}\left(h_{<k}\right) \\
& \leq-\log \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)+\frac{1}{\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E)^{2}} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{k=0}^{t-1} \theta_{q}\left(h_{<k}\right) \tag{A.46}
\end{align*}
$$

where (a) follows from $h_{<t}$ satisfying $E$ with ${ }^{E} \mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}$-prob. 1, (b) follows because $\mu$ and $\pi^{d}$ are fair, and (c) follows from Equation A. 45 and Lemma 13.

Finally,

$$
\begin{align*}
\mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \sum_{k=0}^{t-1} \theta_{q}\left(h_{<k}\right) & =t \mathbb{E}_{k \sim \mathbb{U}([t])} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \theta_{q}\left(h_{<k}\right) \\
& =t\left(\left(\mathbb{E}_{k \sim U([t])} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \theta_{q}\left(h_{<k}\right)\right)^{3}\right)^{1 / 3} \\
& \stackrel{(a)}{\leq} t\left(\mathbb{E}_{k \sim \mathbb{U}([t])} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \theta_{q}\left(h_{<k}\right)^{3}\right)^{1 / 3} \\
& =t\left(\frac{1}{t} \sum_{k=0}^{t-1} \mathbb{E}_{\mu}^{\pi_{\alpha}^{i}} \theta_{q}\left(h_{<k}\right)^{3}\right)^{1 / 3} \\
& \stackrel{(b)}{\leq} t^{2 / 3}|\mathcal{A}|^{1 / 3} \alpha^{-1}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)^{1 / 3} \tag{A.47}
\end{align*}
$$

where (a) follows from Jensen's Inequality, and (b) follows from Theorem 4. Combining this with Inequality A.46, and recalling $\mathrm{P}_{\mu}^{\pi_{\alpha}^{i}}(E) \geq 1-\alpha / w\left(\pi^{d}\right)$, we have

$$
\begin{equation*}
\operatorname{KL}_{t}\left({ }^{E} \mathrm{P}_{\mu}^{\pi_{i}^{i}} \|{ }^{E} \mathrm{P}_{\mu}^{\pi^{d}}\right) \leq \frac{\alpha^{-1}|\mathcal{A}|^{1 / 3}\left(24 w\left(\pi^{d}\right)^{-1}+12\right)^{1 / 3}}{\left(1-\alpha / w\left(\pi^{d}\right)\right)^{2}} t^{2 / 3}-\log \left(1-\alpha / w\left(\pi^{d}\right)\right) \tag{A.48}
\end{equation*}
$$

## A. 7 Correctness of Algorithm 2

In this section, we show that Algorithm 2 performs approximate inversion. And then we show how to modify the algorithm in the setting were all columns of $U$ are identical, for a factor of $q$ speedup.

We begin by writing the exact form of $\left(u^{\prime}\right)^{p=l}$ from the proof of Theorem 11, and
the corresponding ${c^{\prime}}^{(l)}$. Recall the Sherman-Morrison Formula:

$$
\left(A+c u u^{\top}\right)^{-1}=A^{-1}-\frac{A^{-1} u u^{\top} A^{-1}}{c^{-1}+u^{\top} A^{-1} u}
$$

At the time that we update $A$ with the rank one matrix $c^{(l)} u^{p=l} u^{p=l^{\top}}$, what does $A$ equal? Let $c$ and $u$ originate from the $i^{\text {th }}$ column of $C$ and $U$. So our update is $C_{:, i}^{(l)} U_{:, i}^{P_{: i}=l} U_{: ; i}^{P_{i, i}=l^{\top}}$.
Then $A \leftarrow A_{i+1}=I+\sum_{k=i+1}^{q} \sum_{\ell \in[r]} C_{:, k}^{(\ell)} U_{: ;, k}^{P_{:, k}=\ell} U_{:, k}^{P_{i, k}=\ell^{\top}}$, recalling $r$ is the largest integer in $P$. And likewise, all the relevant entries of $C^{\prime}$ and $U^{\prime}$ have been calculated for $A^{-1}$. So we have $A_{i+1}^{-1}=I+\sum_{k=i+1}^{q} \sum_{\ell \in[r]} C_{:, k}^{\prime(\ell)} U_{:, k}^{\prime P_{i, k}=\ell} U^{\prime} P_{:, k}^{P_{i, k}}=\ell^{\top}$. The main computation we need to do is $A^{-1} u$. We'll say that $k_{\ell} \sqsubseteq i_{l}$ if set $l$ from partition $P_{:, i}$ is a superset of set $\ell$ from partition $P_{:, k}$. That is, the rows where $P_{:, k}$ takes the value $\ell$ are a subset of the rows where $P_{:, i}$ takes the value $l$. The key simplification we use is that if $k_{\ell} \nsubseteq i_{l}$, then $U_{: ; i, i}^{P_{i, i}=l^{\top}} U_{: ;, k}^{\prime P_{i, k}=\ell}=0$. The rows at which those two vectors have nonzero elements are disjoint. This logic is explained in the proof of Theorem 11 without all the notation.

So we let

$$
\begin{align*}
& U_{: ; i}^{\prime P_{: i, i}=l}=A_{i+1}^{-1} U_{:, i}^{P_{: i, i}=l}=\left(I+\sum_{k=i+1}^{q} \sum_{\ell \in[r]} C_{:, k}^{\prime(\ell)} U_{:, k}^{\prime P_{:, k}=\ell} U_{:, k}^{\prime P_{:, k}=\ell^{\top}}\right) U_{: ; i}^{P_{: i, i}=l} \\
& =\left(I+\sum_{k=i+1}^{q} \sum_{\ell: k_{\ell} \sqsubseteq i_{l}} C_{:, k}^{\prime(\ell)}{U^{\prime}}_{: P_{:, k}=\ell}^{:=\ell} U_{:, k}^{\prime P_{:, k}=\ell^{\top}}\right) U_{:, i}^{P_{:, i}=l}  \tag{A.49}\\
& =U_{:, i}^{P_{: i, i}=l}+\sum_{k=i+1}^{q} \sum_{\ell: k_{\ell} \sqsubseteq i_{l}} C_{:, k}^{\prime(\ell)} U_{:, k}^{\prime P} P_{:, k}=\ell \quad U_{:, k}^{\prime P_{:, k}=\ell^{\top}} U_{:, i}^{P_{:, k}=\ell} \tag{А.50}
\end{align*}
$$

Equation A. 51 follows because the terms in the dot product ${U_{i, k}^{\prime P}{ }_{i, k}=\ell^{\top}}_{U_{:, i}^{P_{i ; i}=l}}$ are only nonzero when $P_{:, k}=\ell$ and $P_{:, i}=l$, but the latter is implied by $k_{\ell} \sqsubseteq i_{l}$, so this is equivalent to the elements where $P_{:, k}=\ell$. So $U_{:, k}^{P_{: k}, k=\ell^{\top}} U_{:, i, i}^{P_{: i}=l}=$ ${U^{:, k}}_{P_{:, k}=\ell{ }^{\top}}^{U_{:, i}^{P_{i, k}=\ell}}$, which gives us Equation A.51. Then, we let

$$
\begin{equation*}
C_{:, i}^{\prime(l)}=\frac{-1}{1 / C_{:, i}^{(l)}+U_{:, i, i}^{P_{: i}=l^{\top}} U_{:, i}^{P_{; i}=l}} \tag{A.52}
\end{equation*}
$$

Proposition 3 (Correctness of Algorithm 2). In Algorithm 2, $U^{\prime}$ and $C^{\prime}$ take the values defined in Equations A.51 and A.52.

Proof. We'll assume that for $k>i, C^{\prime}{ }_{:, k}$ and $U^{\prime}{ }_{:, k}$ have the right values, and we'll show that $C^{\prime}{ }_{:, i}$ and $U_{i, i}^{\prime}$ get the right values. Starting with $U_{:, q}^{\prime}$, the $\sum_{k=q+1}^{q}$ in Equation A. 51 is empty, so $U_{:, q}^{\prime}=U_{:, q}$. In Algorithm 2, $U^{\prime}$ is initialized to $U$, and the $q^{\text {th }}$ column is never updated.

Now we see that $C^{\prime}{ }_{; i,}$ is correct assuming $U^{\prime}{ }_{:, i}$ is. In Line 5, we ensure $z_{l}=$ $C_{:, i}^{(l)} U_{: ; i}^{\prime P_{i, i}=l^{\top}} U_{:, i}^{P_{i, i}=l}$. A dot product is the sum of elementwise multiplications, and one can inspect that each such multiplication gets added to the right slot in z. Then, Line 6 implements Equation A.52, with numerator and denominator multiplied by $C_{:, i}^{(l)}$. (It stores the same value in multiple locations).

Now we turn to $U_{:, i}^{\prime}$. $U_{:, i}^{\prime}$ is updated in every preceding loop. It starts out initialized to $U_{:, i}$, which accounts for the first term in Equation A.51. In the sum from $k=i+1$ to $q$, each term is accounted for in a separate loop of the algorithm. We check that each term gets added at some point. So consider the term $\sum_{\ell: k_{\ell} \sqsubseteq i_{l}} C_{:, k}^{\prime(\ell)} U_{:, k}^{\prime P_{i, k}=\ell} U_{:, k}^{\prime P_{i, k}=\ell^{\top}} U_{: ;, i}^{P_{i, i}=l}$.

This term gets added to $U_{:, i}$ when $i$ from Algorithm 2 equals $k$ from Equation A.51, and when $k$ from Algorithm 2 equals $i$ from Equation A.51. We are very
sorry about this correspondence, but it would have to happen either here or above
 Then, in Line 11, $y_{p_{j} k}$ gets multiplied by $\left(U^{\prime}\right)_{:, i}^{P_{:, i}=p_{j}}$ and $\left.C_{:, i}^{\prime} p_{j}\right)$, and added to $U_{:, k}^{\prime}$. Swapping the $i$ 's and $k$ 's, and letting $p_{j}$ from Algorithm 2 equal $\ell$ from Equation A.51, Lines 10 and 11 add to $U_{:, k}^{\prime}$ the terms in Equation A. 51 .

Thus, doing induction from $i=q$ down to 1 , Algorithm 2 assigns $C_{:, i}^{\prime}$ and $U_{:, i}^{\prime}$ the correct values.

Now, we modify Algorithm 2 for the setting where all the columns $U_{:, i}$ are the same, allowing a speedup of $O(q)$.

Proposition $4(O(n q)$ inversion). Algorithm 6 performs approximate inversion in $O(n q)$ time.

Proof. The fact that Algorithm 6 runs in $O(n q)$ time is easily verified. Up to Line 12, Algorithm 6 is the same as Algorithm 2, except $U$ has been replaced with $u\left(\mathbf{1}^{q}\right)^{\top}$. So all we have to show is that Lines 13-16 produce the same result that Lines 10-12 would have.

Observe that at the start of the algorithm, $U_{:, k}^{\prime}$ and $U_{:, k+1}^{\prime}$ are initialized to the same value. Observe that Lines 11 and 12 repeat the same computation $i-1$ times. So in Lines 11 and 12, $U_{:, k}^{\prime}$ and $U_{:, k+1}^{\prime}$ are updated by the same amount if $i>k+1$, and they aren't updated at all if $i \leq k$. So the difference between $U_{:, k}^{\prime}$ and $U_{:, k+1}^{\prime}$ comes from only the former being updated when $i=k+1$.

Therefore, Line 14 initializes $U_{:, k}^{\prime}$ to $U_{:, k+1}^{\prime}$. Then Lines 15 and 16 update $U_{:, k}^{\prime}$ with the appropriate difference; they copy Lines 11 and 12 , setting $k$ to $i-1$.

```
Algorithm 6 Inverse and determinant of \(I+\) SROS Linear Operator, in which all
columns of \(U\) are the same.
Require: \(P \in[r]^{n \times q}, C \in \mathbb{R}^{n \times q}, u \in \mathbb{R}^{n}\)
Ensure: \(I+L\left(P, C^{\prime}, U^{\prime}\right)=\left(I+L\left(P, C, u\left(\mathbf{1}^{q}\right)^{\top}\right)\right)^{-1} ; \quad x=\log \mid I+\)
    \(L\left(P, C, u\left(\mathbf{1}^{q}\right)^{\top}\right) \mid\)
    \(x, C^{\prime}, U^{\prime} \leftarrow 0, \mathbf{0} \in \mathbb{R}^{n \times q}, u\left(\mathbf{1}^{q}\right)^{\top}\)
    for \(i \in(q, q-1, \ldots, 1)\) do \(\triangleright O(n q)\) time
        \(p, c, u^{\prime} \leftarrow P_{:, i}, C_{:, i}, U_{:, i}^{\prime}\)
        \(z \leftarrow \mathbf{0} \in \mathbb{R}^{r} \quad \triangleright z_{i}\) will store \(c^{(l)}\left(\left(u^{\prime}\right)^{p=l}\right)^{\top} u^{p=l}\),
        where \(c_{k}=c^{(l)}\) if \(p_{k}=l\)
                                \(\triangleright O(n)\) time
            for \(j \in[n]\) do \(C_{j i}^{\prime} \leftarrow-c_{j} /\left(1+z_{p_{j}}\right)\)
                            \(\triangleright O(n)\) time
            for \(i \in[r]\) do \(x \leftarrow x+\log \left(1+z_{i}\right)\)
                            \(\triangleright O(n)\) time
        if \(\mathrm{i}>0\) then
            if False then \(\triangleright\) This block is the slow version. What follows below is
    equivalent.
            \(y \leftarrow \mathbf{0} \in \mathbb{R}^{n \times i}\)
            for \(j, k \in[n] \times[i-1]\) do \(y_{p_{j} k} \leftarrow y_{p_{j} k}+u_{j}^{\prime} u_{j}\)
            for \(j, k \in[n] \times[i-1]\) do \(U_{j k}^{\prime} \leftarrow U_{j k}^{\prime}+C_{j i}^{\prime} u_{j}^{\prime} y_{p_{j} k}\)
            \(y \leftarrow \mathbf{0} \in \mathbb{R}^{n} \quad \triangleright O(n)\) time
            \(U_{:,(i-1)}^{\prime} \leftarrow U_{:, i}^{\prime} \quad \triangleright O(n)\) time
            for \(j \in[n]\) do \(y_{p_{j}} \leftarrow y_{p_{j}}+u_{j}^{\prime} u_{j} \quad \triangleright O(n)\) time
            for \(j \in[n]\) do \(U_{j(i-1)}^{\prime} \leftarrow U_{j(i-1)}^{\prime}+C_{j i}^{\prime} u_{j}^{\prime} y_{p_{j}} \quad \triangleright O(n)\) time
    return \(C^{\prime}, U^{\prime}, x\)
```


## A. 8 Gradient of loss with respect to weights

In this section, we show how to calculate $\nabla_{w}$ NLL in $O\left(n q^{2}\right)$ time. Recall:

$$
\begin{equation*}
\operatorname{NLL}(w)=\frac{1}{2}\left(y^{\top}\left(K_{X X}(w)+\lambda I\right)^{-1} y+\log \left|K_{X X}(w)+\lambda I\right|+n \log (2 \pi)\right) . \tag{A.53}
\end{equation*}
$$

Differentiating gives:

$$
\begin{equation*}
\frac{\partial \mathrm{NLL}}{\partial w_{i}}=\frac{1}{2}\left[-y^{\top}\left(K_{X X}+\lambda\right)^{-1} \frac{\partial K_{X X}}{\partial w_{i}}\left(K_{X X}+\lambda\right)^{-1} y+\operatorname{Tr}\left(\frac{\partial K_{X X}}{\partial w_{i}}\left(K_{X X}+\lambda\right)^{-1}\right)\right] \tag{A.54}
\end{equation*}
$$

We begin by evaluuating $\partial K_{X X} / \partial w_{i}$. Recall from Proposition 2 that $K_{X X}=$ $L(P, C, U)$, where $C=1^{n} w^{\top}$, and $U=\mathbf{1}^{n \times q}$. It follows easily from the definition of $L$ that the elements of $L(P, C, U)$ are linear in the elements of $C$. So,

$$
\begin{equation*}
\frac{\partial K_{X X}}{\partial w_{i}}=L\left(P_{:, i}, \mathbf{1}^{n}, \mathbf{1}^{n}\right) \tag{A.55}
\end{equation*}
$$

Algorithm 3 shows how to calculate $\left(K_{X X}+\lambda\right)^{-1}$ in $O(n q)$ time, and represent it as $L\left(P, C^{-1}, U^{-1}\right)$. (Recall $C^{-1}$ and $U^{-1}$ are not true inverses; we just the notation to denote their purpose.) Now we turn to the question of how to calculate $\operatorname{Tr}\left[L\left(P_{:, i}, \mathbf{1}^{n}, \mathbf{1}^{n}\right) L\left(P, C^{-1}, U^{-1}\right)\right]$. We are considering symmetric matrices, so the trace of the product is the sum of the elements of the elementwise product. We expand and simplify:

$$
\begin{align*}
T & :=\operatorname{Tr}\left[L\left(P_{:, i}, \mathbf{1}^{n}, \mathbf{1}^{n}\right) L\left(P, C^{-1}, U^{-1}\right)\right]  \tag{A.56}\\
& =\sum_{j=1}^{q} \operatorname{Tr}\left[L\left(P_{:, i}, \mathbf{1}^{n}, \mathbf{1}^{n}\right) L\left(P_{:, j}, C_{:, j}^{-1}, U_{:, j}^{-1}\right)\right]  \tag{A.57}\\
& \stackrel{(a)}{=} \sum_{j=1}^{q} \operatorname{Tr}\left[L\left(P_{:, \max (i, j)}, \mathbf{1}^{n}, \mathbf{1}^{n}\right) L\left(P_{:, \max (i, j)}, C_{:, j}^{-1}, U_{:, j}^{-1}\right)\right]  \tag{А.58}\\
& \stackrel{(b)}{=} \sum_{j=1}^{q} \sum_{\text {elements }} L\left(P_{:, \max (i, j)}, C_{:, j}^{-1}, U_{:, j}^{-1}\right)  \tag{A.59}\\
& \stackrel{(c)}{=} \sum_{j=1}^{q} \sum_{\ell=1}^{\max \left(P_{:, \max (i, j)}\right)} \sum_{\text {elements }}\left(C_{:, j}^{-1}\right)^{(\ell)}\left(U_{:, j}^{-1}\right)^{P_{:, \max }(i, j)}=\ell\left(U_{:, j}^{-1}\right)^{P_{:, \max (i, j)}=\ell^{\top}}  \tag{A.60}\\
& =\sum_{j=1}^{q} \sum_{\ell=1}^{\max \left(P_{i, m a x}(i, j)\right)}\left(C_{:, j}^{-1}\right)^{(\ell)}\left\|\left(U_{:, j}^{-1}\right)^{P_{:, m a x}(i, j)}=\ell\right\|_{1}^{2} \tag{A.61}
\end{align*}
$$

where ( $a$ ) follows from the fact that the $(i, j)^{\text {th }}$ element of $L(p, c, u)$ is zero unless $p_{i}=p_{j}$, in which case, it is $c_{i} u_{i} u_{j}$; when multiplying elementwise by $L\left(p^{\prime}, c^{\prime}, u^{\prime}\right)$, where $p^{\prime}$ is a finer partition, $L(p, c, u) \odot L\left(p^{\prime}, c^{\prime}, u^{\prime}\right)=L\left(p^{\prime}, c, u\right) \odot L\left(p^{\prime}, c^{\prime}, u^{\prime}\right)$, because $L(p, c, u)$ and $L\left(p^{\prime}, c, u\right)$ only differ on elements where $L\left(p^{\prime}, c^{\prime}, u^{\prime}\right)$ is 0 anyway. In the context of $(a), P_{;, \max (i, j)}$ is a finer partition than $P_{;, \min (i, j)}$. (b) follows because we are doing elementwise multiplication between the two matrices; anywhere $L\left(P_{:, \max (i, j)}, \mathbf{1}^{n}, \mathbf{1}^{n}\right)$ is $0, L\left(P_{:, \max (i, j)}, C_{:, j}^{-1}, U_{;, j}^{-1}\right)$ is already 0 , and elsewhere, multiplying elements by 1 does not effect the matrix. (c) follows from the construction of $L$.

It is straightforward to compute this in $O(n q)$ time. See Algorithm 7, which runs in $O(n)$ time and can be iterated over the $q$ terms in Equation A.61.

Now we can see that Equation A. 54 can be computed for all $w_{i}$ in $O\left(n q^{2}\right)$ time. First, $\left(K_{X X}+\lambda\right)^{-1}$ can be computed in $O(n q)$ time, in the form $L\left(P, C^{-1}, U^{-1}\right)$,

```
Algorithm 7 Calculate \(\sum_{\ell=1}^{\max (p)} c^{(\ell)}\left\|u^{p=\ell}\right\|_{1}^{2}\).
Require: \(p \in[r]^{n}, c, u \in \mathbb{R}^{n}\)
Ensure: \(x=\sum_{\ell=1}^{m} c^{(\ell)}\left\|u^{p=\ell}\right\|_{1}^{2}\)
    \(y \leftarrow \mathbf{0}^{m}\)
    for \(j \in[n]\) do \(y_{p_{j}} \leftarrow y_{p_{j}}+\sqrt{c_{j}} u_{j} \quad \triangleright\)
    \(\sqrt{c_{j}}\) may be imaginary, but it will later
    be squared. With modifications, we could
    avoid complex types.
    \(x \leftarrow 0\)
    for \(j \in[r]\) do \(x \leftarrow x+y_{j}^{2}\)
    return \(x\)
```

as shown in Algorithm 3. Then, $z=\left(K_{X X}+\lambda\right)^{-1} y$ can be computed in $O(n q)$ time, also as shown in Algorithm 3. Then, for each $i \in[q]$, we can calculate $-z^{\top} \frac{\partial K_{X X}}{\partial w_{i}} z=-z^{\top} L\left(P_{:, i}, 1^{n}, \mathbf{1}^{n}\right) z$ in $O(n)$ time. Thus, handling the first term in Equation A. 54 for all $i$ takes a total of $O(n q)$ time. As just shown, the second term can be computed in $O(n q)$ time for each $i$, giving a total run time of $O\left(n q^{2}\right)$.

## A. 9 Experimental Details

The following section was mostly drafted by my co-author Sam Daulton.
We initialize 160 random bit orders. For each one, we initialize three weight vectors $w$ : uniform, uniform except the last bit is 0.5 , and uniform except the last bit is 0.9 . Out of these 480 initializations, we draw 20 samples via Boltzman sampling (Duchon et al., 2004) on the log likelihood of the training data (after standardizing the values to have zero mean and unit variance). Then, we optimize the weights and bit order with BFGS as described in Section 5.5, using line search with Wofle conditions, with no extra gradient computations during line search. This allows fewer calculations of the gradient relative to the cheaper calculation of the loss. The BT column in Table 5.1 refers to the performance of the binary
tree kernel, using the weights and bit order that gave the lowest training NLL out of these 20 trained models.

BTE produces a Gaussian mixture model at each predictive location, mixing over the predictive Gaussians produced by each of these 20 trained models. The relative weights of each Gaussian in the mixture depends on the training NLL of the model that produced it. We weight the models according to the softmax of the per-datapoint NLL with a temperature of 0.01 .

We follow the same train/test/validation splits as Wang et al. (2019), but we never use the validation set, which the methods we compare against need. Thus, we could add the validation data to the training data for the binary tree kernel and call it a fair comparison, but we didn't do this, so as not to confuse the origin of the binary tree kernel's success.

## A. 10 Additional Empirical Evaluation

## A.10.1 Sensitivity Analysis on Precision $p$

In Table A.1, we evaluate the performance of the BT and BTE kernels on the precision $p$ for $p \in(2,4,8)$. In all problems, we find that RMSE and NLL decrease monotonically as $p$ increases and wall time increases monotonically. For best predictive performance, $p$ should be set as large as possible subject resource constraints. This validates that our heuristic rule for setting $p$ is a reasonable choice in a variety of settings.

## A.10.2 A Simple Performance Improvement

Much of the following section was written by my co-author Sam Daulton.

| Dataset | $n$ | d | BTE ( $p=2$ ) | BTE $(p=4)$ | BTE ( $p=8$ ) | BT ( $p=2$ ) | BT ( $p=4$ ) | BT ( $p=8$ ) |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Poletele | 9,600 | 26 | $0.772 \pm 0.022$ | $-0.266 \pm 0.023$ | $-0.664 \pm 0.052$ | $0.771 \pm 0.021$ | $-0.198 \pm 0.012$ | $-0.398 \pm 0.159$ |
| Elevators | 10,623 | 18 | $1.095 \pm 0.046$ | $0.761 \pm 0.024$ | $0.654 \pm 0.023$ | $1.097 \pm 0.049$ | $0.756 \pm 0.019$ | $0.662 \pm 0.018$ |
| Bike | 11,122 | 17 | $1.095 \pm 0.046$ | $0.761 \pm 0.024$ | $0.654 \pm 0.023$ | $0.583 \pm 0.013$ | $-0.004 \pm 0.020$ | $-0.800 \pm 0.271$ |
| Kin40k | 25,600 | 8 | $0.888 \pm 0.004$ | $0.871 \pm 0.009$ | $0.869 \pm 0.004$ | $0.894 \pm 0.005$ | $0.879 \pm 0.012$ | $0.882 \pm 0.006$ |
| Protein | 29,267 | 9 | $1.280 \pm 0.007$ | $1.042 \pm 0.012$ | $0.781 \pm 0.022$ | $1.281 \pm 0.007$ | $1.048 \pm 0.013$ | $0.842 \pm 0.032$ |
| KeggDir | 31,248 | 20 | $0.917 \pm 0.028$ | $-0.607 \pm 0.019$ | $-1.030 \pm 0.019$ | $0.916 \pm 0.029$ | $-0.608 \pm 0.017$ | $-1.028 \pm 0.023$ |
| CTslice | 34,240 | 385 |  |  |  |  |  |  |
| KEGGU | 40,708 | 27 | $0.228 \pm 0.057$ | $-0.607 \pm 0.009$ | $-0.668 \pm 0.007$ | $0.228 \pm 0.058$ | $-0.606 \pm 0.008$ | $-0.677 \pm 0.015$ |
| 3DRoad | 278,319 | 3 | $1.292 \pm 0.007$ | $0.973 \pm 0.007$ | $-0.255 \pm 0.004$ | $1.295 \pm 0.003$ | $0.981 \pm 0.004$ | $-0.251 \pm 0.005$ |
| Song | 329,820 | 90 | $1.328 \pm 0.001$ | - |  | $1.317 \pm 0.014$ |  |  |
| Buzz | 373,280 | 77 | $1.198 \pm 0.003$ | $1.107 \pm 0.009$ |  | $1.198 \pm 0.003$ | $1.106 \pm 0.009$ |  |
| HouseElec | 1,311,539 | 11 | $0.629 \pm 0.003$ | $-0.673 \pm 0.003$ | $-2.569 \pm 0.006$ | $0.629 \pm 0.003$ | $-0.669 \pm 0.007$ | $-2.492 \pm 0.012$ |
| Poletele | 9,600 | 26 | $0.513 \pm 0.012$ | $0.185 \pm 0.006$ | $0.159 \pm 0.004$ | $0.514 \pm 0.012$ | $0.194 \pm 0.003$ | $0.160 \pm 0.009$ |
| Elevators | 10,623 | 18 | $0.725 \pm 0.031$ | $0.525 \pm 0.014$ | $0.481 \pm 0.016$ | $0.725 \pm 0.032$ | $0.520 \pm 0.013$ | $0.483 \pm 0.015$ |
| Bike | 11,122 | 17 | $0.430 \pm 0.005$ | $0.237 \pm 0.005$ | $0.120 \pm 0.057$ | $0.431 \pm 0.006$ | $0.237 \pm 0.005$ | $0.104 \pm 0.029$ |
| Kin40k | 25,600 | 8 | $0.590 \pm 0.003$ | $0.582 \pm 0.005$ | $0.580 \pm 0.003$ | $0.593 \pm 0.004$ | $0.586 \pm 0.006$ | $0.587 \pm 0.005$ |
| Protein | 29,267 | 9 | $0.870 \pm 0.006$ | $0.687 \pm 0.010$ | $0.609 \pm 0.008$ | $0.870 \pm 0.006$ | $0.691 \pm 0.010$ | $0.623 \pm 0.010$ |
| Keggidr | 31,248 | 20 | $0.604 \pm 0.017$ | $0.128 \pm 0.003$ | $0.086 \pm 0.003$ | $0.604 \pm 0.017$ | $0.128 \pm 0.003$ | $0.087 \pm 0.003$ |
| CTslice | 34,240 | 385 | - | - | - | - | - | - |
| KEGGU | 40,708 | 27 | $0.302 \pm 0.018$ | $0.128 \pm 0.002$ | $0.120 \pm 0.002$ | $0.302 \pm 0.018$ | $0.129 \pm 0.001$ | $0.119 \pm 0.002$ |
| 3DROAD | 278,319 | 3 | $0.882 \pm 0.005$ | $0.642 \pm 0.004$ | $0.187 \pm 0.000$ | $0.883 \pm 0.003$ | $0.645 \pm 0.002$ | $0.186 \pm 0.001$ |
| Song | 329,820 | 90 | $0.914 \pm 0.001$ | - | - | $0.904 \pm 0.012$ |  |  |
| Buzz | 373,280 | 77 | $0.801 \pm 0.002$ | $0.729 \pm 0.007$ | - | $0.801 \pm 0.002$ | $0.730 \pm 0.007$ | - |
| HouseElec | 1,311,539 | 11 | $0.453 \pm 0.001$ | $0.121 \pm 0.001$ | $0.029 \pm 0.001$ | $0.453 \pm 0.001$ | $0.120 \pm 0.001$ | $0.029 \pm 0.001$ |
| PoleTele | 9,600 | 26 | $0.600 \pm 0.000$ | $2.500 \pm 0.200$ | $8.600 \pm 0.600$ | $0.600 \pm 0.000$ | $2.500 \pm 0.200$ | $8.600 \pm 0.600$ |
| Elevators | 10,623 | 18 | $0.500 \pm 0.100$ | $1.600 \pm 0.100$ | $3.000 \pm 0.400$ | $0.500 \pm 0.100$ | $1.600 \pm 0.100$ | $3.000 \pm 0.400$ |
| Bike | 11,122 | 17 | $0.400 \pm 0.000$ | $1.700 \pm 0.100$ | $3.100 \pm 0.200$ | $0.400 \pm 0.000$ | $1.700 \pm 0.100$ | $3.100 \pm 0.200$ |
| Kin40k | 25,600 | 8 | $0.300 \pm 0.000$ | $1.200 \pm 0.200$ | $1.500 \pm 0.000$ | $0.300 \pm 0.000$ | $1.200 \pm 0.200$ | $1.500 \pm 0.000$ |
| Protein | 29,267 | 9 | $0.200 \pm 0.000$ | $0.600 \pm 0.000$ | $2.800 \pm 0.100$ | $0.200 \pm 0.000$ | $0.600 \pm 0.000$ | $2.800 \pm 0.100$ |
| KeggDir | 31,248 | 20 | $0.400 \pm 0.000$ | $2.300 \pm 0.100$ | $7.800 \pm 0.600$ | $0.400 \pm 0.000$ | $2.300 \pm 0.100$ | $7.800 \pm 0.600$ |
| CTslice | 34,240 | 385 | - | - | - | - | - | - |
| KEGGU | 40,708 | 27 | $0.800 \pm 0.100$ | $5.900 \pm 0.800$ | $12.400 \pm 1.200$ | $0.800 \pm 0.100$ | $5.900 \pm 0.800$ | $12.400 \pm 1.200$ |
| 3DRoad | 278,319 | 3 | $0.100 \pm 0.000$ | $0.200 \pm 0.000$ | $2.100 \pm 0.100$ | $0.100 \pm 0.000$ | $0.200 \pm 0.000$ | $2.100 \pm 0.100$ |
| Song | 329,820 | 90 | $11.6 \pm 0.8$ | - | - | $11.6 \pm 0.8$ | - | - |
| Buzz | 373,280 | 77 | $20.200 \pm 6.600$ | $54.900 \pm 13.600$ | - | $20.200 \pm 6.600$ | $54.900 \pm 13.600$ | - |
| HouseElec | 1,311,539 | 11 | $3.300 \pm 0.200$ | $37.800 \pm 2.300$ | $118.41 \pm 3.93$ | $3.300 \pm 0.200$ | $37.800 \pm 2.300$ | $118.41 \pm 3.93$ |

Table A.1: A sensitivity analysis of the performance of the BT kernel with respect to the precision $p$. NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM2-16GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). Omitted results were not run due to limited GPU memory.

As shown in Table A.2, the heuristic rule for setting $p$ results in bit strings that preserve the uniqueness of the raw training set for most datasets. However, for the 3dRoad, Song, and Buzz datasets, the percentage of unique bit strings is very low relative to the percentage of unique training rows. We note that this observation is from exploratory data analysis (EDA) and can be made before model fitting.

There are two key ways that mapping training rows to bit strings can lower the percent of unique examples. First, a low percentage of unique bit strings can arise if a given input feature has a non-uniform input distribution which can lead to multiple different inputs mapping to the same discrete bucket. To alleviate, this problem EDA can be used to determine a suitable feature transformation. For example, we apply a strictly increasing, piecewise linear transformation to the data, mapping the $k^{\text {th }}$ percentile of each dimension to $k / 100$, for $k \in\{0,10,20, \ldots, 100\}$. This resembles an empirical cumulative density function (ECDF). Second, if the precision $p$ is set too low, then multiple different inputs can map to the same bit string. A simple solution is to iteratively increase the precision $p$ based on the difference between the percentage of unique rows in the raw training set and unique bit strings under precision $p$. The last column of Table A. 2 reports the percentage of unique bit strings on the 3dRoad, Song, and Buzz datasets after applying transforming each feature through its ECDF and increasing the precision if need be. These changes (all made through EDA) lead to significantly more unique bit strings. Table A. 3 shows the performance on these datasets under the proposed ECDF transformations and settings of $p$. We find that the BTE outperforms all methods on 3dRoad and Buzz with respect to RSME and NLL under these proposed changes. The performance on Song also improves.

| Dataset | $n$ | $d$ | $p$ | \% Unique Training Rows | \% Unique Bit Strings under $p$ | ... After Transform \& w/ $p_{\text {new }}$ | $p_{\text {NEW }}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| PoleTele | 9,600 | 26 | 6 | 99.9 | 98.8 |  |  |
| Elevators | 10,623 | 18 | 8 | 100.0 | 100.0 |  |  |
| Bike | 11,122 | 17 | 8 | 100.0 | 100.0 |  |  |
| Kin40K | 25,600 | 8 | 8 | 100.0 | 100.0 |  |  |
| Protein | 29,267 | 9 | 8 | 97.4 | 96.5 |  |  |
| KeggDir | 31,248 | 20 | 8 | 36.1 | 36.1 |  |  |
| CTslice | 34,240 | 385 | 1 | 99.9 | 95.5 |  |  |
| KEGGU | 40,708 | 27 | 6 | 32.5 | 32.3 |  |  |
| 3DRoad | 278,319 | 3 | 8 | 99.3 | 15.9 | 97.7 | 16 |
| Song | 329,820 | 90 | 2 | 100.0 | 46.2 | 100 | 2 |
| Buzz | 373,280 | 77 | 2 | 98.5 | 0.5 | 94.6 | 3 |
| HouseElec | 1,311,539 | 11 | 8 | 100.0 | 100.0 |  |  |

Table A.2: Percentage of unique training inputs (over all training inputs) and bit strings (over all training inputs) under the precision $p$ set according to the heuristic rule. We reported the means across 3 training, validation, test set partitions. The last column shows percentage of unique bit strings after transforming each feature through its ECDF.

| DATASET | $n$ | $d$ | BTE | BT | MATÉRN (BBMM) | SGPR | SVGP |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 3DROAD | 278,319 | 3 | $\mathbf{- 1 . 2 8 5} \pm 0.008$ | $-1.267 \pm 0.005$ | $0.909 \pm 0.001$ | $0.943 \pm 0.002$ | $0.697 \pm 0.002$ |
| SONG | 329,820 | 90 | $1.306 \pm 0.011$ | $1.331 \pm 0.003$ | $\mathbf{1 . 2 0 6} \pm 0.024$ | $1.213 \pm 0.003$ | $1.417 \pm 0.000$ |
| BUZZ | 373,280 | 77 | $\mathbf{0 . 0 1 7} \pm 0.002$ | $0.034 \pm 0.000$ | $0.267 \pm 0.028$ | $0.106 \pm 0.008$ | $0.224 \pm 0.050$ |
| 3DROAD | 278,319 | 3 | $\mathbf{0 . 1 0 4} \pm 0.002$ | $\mathbf{0 . 1 0 5} \pm 0.002$ | $\mathbf{0 . 1 0 1} \pm 0.007$ | $0.661 \pm 0.010$ | $0.481 \pm 0.002$ |
| SONG | 329,820 | 90 | $0.894 \pm 0.010$ | $0.904 \pm 0.012$ | $\mathbf{0 . 8 0 7} \pm 0.024$ | $\mathbf{0 . 8 0 3} \pm 0.002$ | $0.998 \pm 0.000$ |
| BUZZ | 373,280 | 77 | $\mathbf{0 . 2 4 9} \pm 0.001$ | $0.253 \pm 0.000$ | $\mathbf{0 . 2 8 8} \pm 0.018$ | $0.300 \pm 0.004$ | $0.304 \pm 0.012$ |
| 3DROAD | 278,319 | 3 | $14.2 \pm 0.2$ | $126.37^{*} \pm 20.92^{*}$ | $12.01 \pm 5.51$ | $34.09 \pm 3.19$ |  |
| SONG | 329,820 | 90 | $11.5 \pm 0.7$ | $33.79^{*} \pm 10.45^{*}$ | $\mathbf{7 . 8 9} \pm 3.12$ | $39.55 \pm 3.08$ |  |
| BUZZ | 373,280 | 77 |  | $82.8 \pm 5.7$ | $571.15^{*} \pm 66.34^{*}$ | $\mathbf{2 9 . 2 5} \pm 18.33$ | $46.35 \pm 2.93$ |

Table A.3: NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM2-16GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods) after transforming each feature through its ECDF and using precision $p_{\text {new }}$. The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU.

## A.10.3 Comparison with Simplex-GP

Here, we compare against Simplex-GP (Kapoor et al., 2021) using the 4 datasets that are common to both papers. We find that BTE outperforms Simplex GP on all datasets with respect to NLL and on 3 out of 4 datasets with respect to RMSE. Furthermore, Simplex-GP is not the top performing method on any dataset.

| DATASET | $n$ | $d$ | BTE | BT | MATÉRN (BBMM) | SGPR | SVGP |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| ELEVATORS | 10,623 | 18 | $0.649 \pm 0.032$ | $0.646 \pm 0.023$ | $0.619 \pm 0.054$ | $0.580 \pm 0.060$ | $\mathbf{0 . 5 1 9} \pm 0.022$ |  |
| PROTEIN | 29,267 | 9 | $\mathbf{0 . 7 8 1} \pm 0.023$ | $0.845 \pm 0.026$ | $1.018 \pm 0.056$ | $0.970 \pm 0.010$ | $1.035 \pm 0.006$ | $1.406 \pm 0.020$ |
| KEGGDDR | 31,248 | 20 | $-1.031 \pm 0.020$ | $-1.029 \pm 0.021$ | $-0.199 \pm 0.381$ | $-\mathbf{1 . 1 2 3} \pm 0.016$ | $-0.940 \pm 0.020$ | $0.797 \pm 0.031$ |
| HouseElec | $1,311,539$ | 11 | $\mathbf{- 2 . 5 6 9} \pm 0.006$ | $-2.492 \pm 0.012$ | $-0.152 \pm 0.001$ | - | $-1.010 \pm 0.039$ | $0.756 \pm 0.075$ |
| ELEVATORS | 10,623 | 18 | $0.478 \pm 0.021$ | $0.476 \pm 0.018$ | $\mathbf{0 . 3 9 4} \pm 0.006$ | $0.437 \pm 0.018$ | $\mathbf{0 . 3 9 9} \pm 0.009$ | $0.510 \pm 0.018$ |
| PROTEIN | 29,267 | 9 | $0.608 \pm 0.008$ | $0.623 \pm 0.011$ | $\mathbf{0 . 5 3 6} \pm 0.012$ | $0.656 \pm 0.010$ | $0.668 \pm 0.005$ | $0.571 \pm 0.003$ |
| KEGGDIR | 31,248 | 20 | $\mathbf{0 . 0 8 6} \pm 0.003$ | $\mathbf{0 . 0 8 6} \pm 0.003$ | $\mathbf{0 . 0 8 6} \pm 0.005$ | $0.104 \pm 0.003$ | $0.096 \pm 0.001$ | $0.095 \pm 0.002$ |
| HouSEELEC | $1,311,539$ | 11 | $\mathbf{0 . 0 2 9} \pm 0.001$ | $\mathbf{0 . 0 2 9} \pm 0.001$ | $0.055 \pm 0.000$ | - | $0.084 \pm 0.005$ | $0.079 \pm 0.002$ |

Table A.4: NLL (top), RMSE (middle), and run time in minutes (bottom) on regression datasets, using a single GPU (Tesla V100-SXM2-16GB for BT and BTE and Tesla V100-SXM2-32GB for the other methods). The asterisk indicates an estimate of the time from the reported training time on 8 GPUS, assuming linear speedup in number of GPUs and independent noise in training times per GPU. All columns except BT and BTE come from Wang et al. (2019) and Simplex-GP results come from Kapoor et al. (2021).


[^0]:    ${ }^{1}$ The conjugate gradients method takes $O\left(n^{2} \sqrt{\kappa}\right)$ time, where $\kappa$ is the condition number of the kernel matrix. Poggio et al. (2019) say "claims about the condition number of a random matrix A should also apply to kernel matrices with random data." If they mean a Wishart random matrix (which it should be if, e.g., $k\left(x, x^{\prime}\right)=x^{\top} x^{\prime}$ ), that would be the square of the condition number of the corresponding Gaussian random matrix, which grows as $O(n)$ (Chen \& Dongarra, 2005). Putting it all together, we get $O\left(n^{3}\right)$ for conjugate gradients. We don't know how quickly preconditioning can reduce the condition number.

