Balancing Exploration and Exploitation: Task-Targeted Exploration for Scientific Decision-Making

by

Genevieve Elaine Flaspohler

B.S.E., University of Michigan (2016)
S.M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science
and to the Joint Program in Applied Ocean Science and Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2022

Author: ____________________

Department of Electrical Engineering and Computer Science
August 19, 2022

Certified by: ____________________

John W. Fisher III
Senior Research Scientist of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: ____________________

Nicholas Roy
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: ____________________

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Accepted by: ____________________

David Ralston
Associate Scientist of Applied Ocean Physics and Engineering
Chair, Joint Committee for Applied Ocean Science and Engineering

Balancing Exploration and Exploitation:
Task-Targeted Exploration for Scientific Decision-Making

by

Genevieve Elaine Flaspohler

Submitted to the Department of Electrical Engineering and Computer Science
and to the Joint Program in Applied Ocean Science and Engineering
on August 19, 2022
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Abstract

How do we collect observational data that reveal fundamental properties of scientific phenomena? This is a key challenge in modern scientific discovery. Scientific phenomena are complex—they have high-dimensional and continuous state, exhibit chaotic dynamics, and generate noisy sensor observations. Additionally, scientific experimentation often requires significant time, money, and human effort. In the face of these challenges, we propose to leverage autonomous decision-making to augment and accelerate human scientific discovery.

Autonomous decision-making in scientific domains faces an important and classical challenge: balancing exploration and exploitation when making decisions under uncertainty. This thesis argues that efficient decision-making in real-world, scientific domains requires *task-targeted exploration*—exploration strategies that are tuned to a specific task. By quantifying the change in task performance due to exploratory actions, we enable decision-makers that can contend with highly uncertain real-world environments, performing exploration parsimoniously to improve task performance.

The thesis presents three novel paradigms for task-targeted exploration that are motivated by and applied to real-world scientific problems. We first consider exploration in partially observable Markov decision processes (POMDPs) and present two novel planners that leverage task-driven information measures to balance exploration and exploitation. These planners drive robots in simulation and oceanographic field trials to robustly identify plume sources and track targets with stochastic dynamics. We next consider the exploration-exploitation trade-off in online learning paradigms, a robust alternative to POMDPs when the environment is adversarial or difficult to model. We present novel online learning algorithms that balance exploitative and exploratory plays optimally under real-world constraints, including delayed feedback, partial predictability, and short regret horizons.

We use these algorithms to perform model selection for subseasonal temperature and precipitation forecasting, achieving state-of-the-art forecasting accuracy.

The human scientific endeavor is poised to benefit from our emerging capacity to integrate observational data into the process of model development and validation. Realizing the full potential of these data requires autonomous decision-makers that can contend with the inherent uncertainty of real-world scientific domains. This thesis highlights the critical role that task-targeted exploration plays in efficient scientific decision-making and proposes three novel methods to achieve task-targeted exploration in real-world oceanographic and climate science applications.

Thesis Supervisor: John W. Fisher III
Title: Senior Research Scientist of Electrical Engineering and Computer Science

Thesis Supervisor: Nicholas Roy
Title: Professor of Aeronautics and Astronautics

Acknowledgements

When you want to build a ship, do not begin by gathering wood, cutting boards,and distributing work, but awaken within the heart of man the desire for the vast and endless sea.

- Antoine de Saint-Exupery

I had the immense good fortune of having double the standard allocation of research advisors during my PhD. I feel tremendous gratitude for John Fisher: I don't know whether to thank you first for your keen technical insights or your keen capacity to use wit and humor to make me laugh. Both were invaluable during the PhD. Thank you to Nicholas Roy for all the ways that you invested in mentorship. Beyond your valuable technical guidance, you taught me how to communicate my science. This is a gift I will forever be grateful for.

Thank you to my committee, Lester Mackey, Claudia Cenedese, and Munther Dahleh: you are the basis vectors for such different and wonderful professional and personal spaces, from oceanography to statistics, from the theoretical to the applied. I benefited tremendously from having such different and thoughtful researchers in my life long before I started writing this thesis. And thank you to the three intrepid fellow students who read this thesis from start to finish: Victoria Preston, Katherine Liu, and David Forman.

My PhD often came in twos. Beyond being co-advised, I was part of two incredible labs, two institutions, and two research collaborations. Like the ship of Theseus, the rotating cast of characters changed during the past six years but the deep sense of community and challenge was a constant structure. From SLI, thank you to David Hayden for showing me how to build a community with intention, always being excited to debate a topic or brainstorm on a whiteboard, and for introducing birthday wisdoms. Thank you to Sue Zheng for generously sharing wisdom and making me stop and listen. Thanks to Stephen Lee for dreaming with me; Christopher Dean for bringing laughter and coffee into our office; Rujian Chen for thoughtful consideration and lessons on the inner lives of hamsters. From the RRG, thank you to Victoria Preston for being my closest comrade in arms, taking on the challenges of marine science with me and teaching me about inclusive and thoughtful leadership. Thank you to Katherine Liu for cathartic conversations over the pottery wheel and elsewhere; to Greg Stein for helping me learn communication in all its forms. A warm thank you to RRG members past: Will Vega-Brown, Kyel Ok, Charlie Guan, Naomi Schurr, Nick Green, Jake Ware, John Carter—you wore the heavy mantle of senior grad student with ease and grace when I joined the lab and I benefited tremendously from your influence. And to RRG members present: Chris Bradley, Mike Noseworthy, Martina Stadler—thank you for growing with me and sharing your lives, porches, dogs, and music taste generously.

I'm grateful for many collaborators across institutions. Thank you to folks at WHOI, especially those who made my first oceanographic expedition possible: the RR2107 research cruise to Guaymas Basin. A special thank you to the Sentry team and the crew of the RV Roger Revelle for allowing me to see my PhD work move a robot in the world. That was a powerful and awe-inspiring moment, built on a well-oiled Rube Goldberg machine of operations. A heartfelt thank you to my subseasonal forecasting comrades—Lester Mackey, Soukayna Mouatadid, Paulo Orenstein, Maria Geogdzhayeva, Judah Cohen, and Ernest Fraenkel. And to my new community at nLine, for inspiring me and casting my gaze forward as I worked to cross this finish line.

Outside of research, I feel fortunate for all the people I encountered in graduate school. Thank you to GW6 and the weekly morning breakfasts led by Leslie and Janet. I first found community and friendship at MIT in those early hours over bagels and fruit plates. Thank you to my cohort of EECS students; I'm not sure what I expected to find in the PhD program, but I was continuously amazed by the kindness, quirkiness, and drive of my fellow students. Thank you to the MIT Women's Volleyball Club —and to Tony and Max, our benevolent leaders—for years of being a physical outlet and a supportive team. And thank you to the EECS Communication Lab, for taking a budding science writer and providing fertile soil and a direction to grow.

This final piece of gratitude runs deep. I feel so blessed by the place that I grew up and the people who grew up with me, from my childhood in the Upper Peninsula to the formative years of college in Ann Arbor. Michigan was my home for most of my life and a place of comfort and support each time the PhD challenged me. My parents, Carrie Vander Veen Flaspohler and David Flaspohler have bathed me in support and belief for as long as I can remember, and they are the reason I have the courage to attempt all that I have. I often feel like my sister Ingrid and my brother Erik have half of my heart in each of their chests; I don't know what I would do without you two. My sister-in-all-but-name, Cecilia Wallace, grew up down the street from me and moved back down the street in time to be a voice of loving support right when I needed it most. The eclectic "potluck group" supported me with everything from moonlight luminary hikes at a low point in my degree to a comfortable work chair to finish out my writing. To my Ann Arbor community—Nicholas Kalweit, Chhavi Chaudhary, Katie Bartek, Josh Adkins, Paulina Devlin, and Jean Carlos Torres Irizarry—you are better and richer friends than I could have hoped for. I was so lucky to find you all and I hope I can carry and deserve your friendship throughout my life.

Finally, I dedicate this thesis to my grandparents, Helen and Ron Flaspohler, who forged a path to the PhD and believed in me endlessly. I love and miss you both.

Contents

List of Symbols

The following tables summarize the symbols used in this thesis, expanded from the notational conventions of [23].

The basics

$\boldsymbol{x}$	A vector $\boldsymbol{x}$.
$\mathbf{X}$	A matrix $\mathbf{X}$.
$\mathbb{X}$	A set $\mathbb{X}$.
X	A random variable X.
x	A constant scalar x.

Specific sets

$\mathbb{R}$	Real numbers.
$\mathbb{R}^n$	Real n-vectors.
$\mathbb{R}^{m \times n}$	Real $m \times n$ matrices.
$\mathbb{R}_+$	Nonegative reals.
$\mathbb{Z}$	Integers.
$\mathbb{Z}_+$	Nonnegative integers.
$\mathbb{S}^n$	Symmetric $n \times n$ matrices.
$\mathbb{S}^n_+$	Symmetric, positive semidefinite $n \times n$ matrices.

Vectors and matrices

$\mathbf{1}$	Vector of all ones.
$\mathbf{e}_i$	ith standard basis vector.
$\mathbf{I}$	The identity matrix.
$\mathbf{X}^\top$	Transpose of matrix $\mathbf{X}$.
$\operatorname{tr} \mathbf{X}$	Trace of matrix $\mathbf{X}$.
$\operatorname{diag} \boldsymbol{x}$	Diagonal matrix with entries $\boldsymbol{x}$.
$\lambda_i(\mathbf{X})$	The ith largest eigenvalue of matrix $\mathbf{X}$.

Norms and distances

$\|\cdot\|$	A norm.
$\|\cdot\|_*$	Dual of norm $\|\cdot\|$.
$\|\boldsymbol{x}\|_2$	The ℓ_2-norm of vector $\boldsymbol{x}$.
$\|\boldsymbol{x}\|_1$	The ℓ_1-norm of vector $\boldsymbol{x}$.
$\|\boldsymbol{x}\|_\infty$	The ℓ_∞-norm of vector $\boldsymbol{x}$.
$\|\mathbf{X}\|_{\mathrm{op}}$	The operator-norm of matrix $\mathbf{X}$.

Probability

$\mathbb{E}_{x \sim X}[f(x)]$	Expected value of $f(x)$ w.r.t. the law of random variable X.
$\mathbb{P}(S)$	Probability of event S or probability distribution of random variable S.
$\triangle_n$	The $(n-1)$-dimensional probability simplex.
$\mathcal{P}(\cdot)$	The set of probability distributions on the input.

Functions and derivatives

$f : A \to B$	f is a function on the set A into the set B.
$C^0(A)$	The set of continuous functions on the set A.
∇f	Gradient of the function f.
$\frac{\partial f}{\partial x}$	The partial derivative of the function f w.r.t. input variable x.

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Our ability to measure the world around us—from the micro- to the macro-scale—has increased rapidly in recent decades. Scientific discovery in fields ranging from the climate sciences to experimental physics is poised to benefit from this increased availability of observational data, which enables inductive reasoning and model development (Fig. 1.1). A key challenge in scientific domains is that observational data are often expensive to obtain in terms of money, time, and human labor. These constraints dictate that a scientist should carefully choose experiments that produce informative measurements and target *prerogative instances*: experiments that are “specially valuable because they enable us to decide between two theories, each possible so far as previous observations are concerned” [151].

Historically, scientific data collection has largely been carried out by individual scientists, who leverage domain knowledge and intuition to design informative experiments and explore complex phenomena. In modern applications, however, scientific phenomena are represented by a high dimensional set of tightly coupled state variables, generate incomplete and noisy observations, and exhibit complex dynamics in space and time. State uncertainty is often significant, and many observations may measure only nuisance variables or provide redundant information. Additionally, a single set of experiments may be insufficient to understand a complex phenomenon; multiple rounds of iterative experimentation and observation must be used to incrementally learn models and reduce parameter uncertainty. In the presence of these complexities, human-driven scientific data collection is often insufficient or inefficient.

As we seek to understand increasingly complex scientific phenomena, there is an opportunity to leverage autonomous decision-making algorithms to augment the human data collection process and extract insight from observational data in new ways. In contrast with a human-driven, “single-shot” experimental design process, this thesis frames scientific data collection as an autonomous, closed-loop and model-based

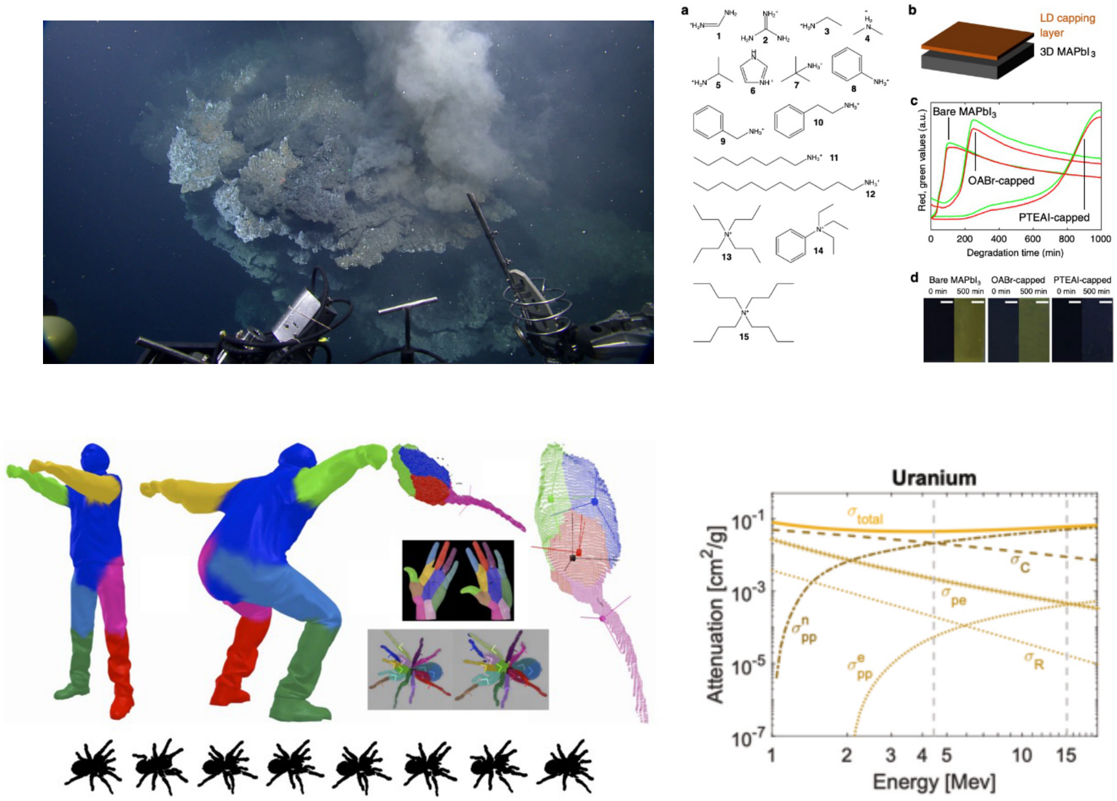


Figure 1.1: ***Applications of Autonomous Scientific Discovery:*** *In many scientific domains, researchers use structured models in combination with artificial intelligence methods to drive scientific discovery. Clockwise from the top-left: exploring hydrothermal plumes with deep sea robots from the WHOI JASON team; optimizing photovoltaic material synthesis from Hartono et al. [55]; dynamic parts modeling for animal behavior studies from Hayden et al. [58]; nuclear cargo detection with physics-based Bayesian models from Zheng [198].*

decision-making problem (Fig. 1.2). The decision-making loop begins with an autonomous planner, which uses available prior domain information to select an initial experiment or set of experiments aimed at exploring some scientific hypothesis. After executing the experiments, the decision-maker receives the resulting observational data. These data are used to refine a predictive model of the scientific phenomena of interest. This may involve tuning the parameters of a partial differential equation or learning the weights of a neural network. The updated model is then available to the planner, which leverages improved model predictions to target the next set of experiments. This decision-making loop occurs

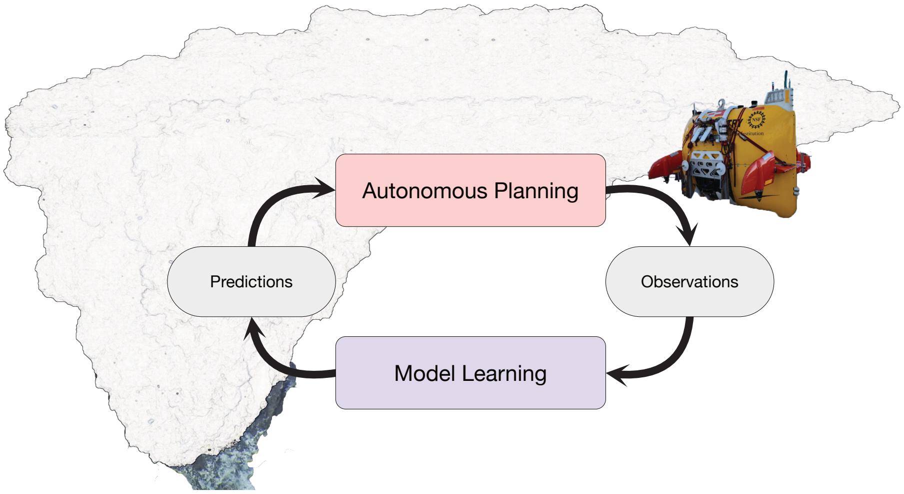


Figure 1.2: ***Closed-loop, Model-based Scientific Decision-Making:*** *The model-based scientific decision-making loop starts with an autonomous planning algorithm that produces a plan for data acquisition. These plans specify, e.g., a data-collection policy for a scientific robot — such as the autonomous underwater vehicle (AUV) SENTRY pictured — or a sequence of configurations for a static sensor. When executed, this plan produces observations of an underlying scientific phenomena, such as the hydrothermal plume pictured. These observations are integrated into a modeling pipeline, in which a predictive model is learned from scientific data. The learned model produces state estimates and performs uncertainty quantification. State predictions or forecasts are fed into the next round of autonomous planning to improve the quality of the collected data.*

iteratively and in closed-loop, i.e., the planner observes the result of the current experiment before choosing experiments in the subsequent stage.

1.1 Scientific Tasks, Exploration, and Exploitation

Posing scientific discovery as a sequential decision-making problem requires the specification of a scientific task. Often, this scientific task is distinct from building an accurate predictive model of the full physical system, which may be unnecessary or infeasible. Instead, in scientific domains, the task is often to target observations or samples that have specific properties. For example, a marine robot may be tasked to collect methane-rich samples in a hydrothermal plume (Fig. 1.2) or capture camera images of a specific species of coral in a reef environment. Accomplishing these tasks does require constructing a model of the environment. However, the robot does not require high-fidelity predictions for low-methane regions of a plume or sandy regions of a reef to collect the desired observations. Not all uncertainty is detrimental and the best model for decision-making is not necessarily the most accurate model; decision-makers only require models that are sufficient for the task at hand. A finite number of observations are available both for learning a predictive model of the environment and for targeting a scientific task, and optimizing the accuracy of the model is often in direct conflict with targeting the desired scientific observations.

This conflict between model improvement and targeting a scientific task is an instantiation of a fundamental challenge in decision-making: balancing exploration and exploitation when making decisions under uncertainty. Faced with uncertainty, a decision-maker must gather information about the structure and dynamics of the world; this is known as *exploration*. However, while exploring, a decision-maker may be forgoing immediate rewards and is unable to perform actions that *exploit* the current knowledge of the environment. This inherent tension between novelty and information gathering, and actions which are known to be good is known as the explore-exploit dilemma [138].

The explore-exploit dilemma is something that human decision-makers contend with on a daily basis and underpins much of strategy, risk-management and decision-theory. The dilemma emerges in a more explicit way for computational decision-makers, such as the closed-loop system pictured in Fig. 1.2. The system can only gather a finite number of observations, and has significant uncertainty about the state and future evolution of the environment. Some actions available to the decision-maker will result in observations that reduce state uncertainty and enable model learning; these are the exploratory actions. Other actions will result in observations that useful for the scientific task at hand; these are exploitative actions. These two objectives are often in direct competition and a key challenge in decision-making under uncertainty is achieving an efficient balance of exploration and exploitation by performing the minimum amount of exploration necessary to ensure an acceptable level of task performance.

1.2 POMDPs and the Challenges of Scientific Decision-Making

Uncertainty is inherent in real-world decision-making; balancing exploration and exploitation is a key challenge in these domains. Partially observable Markov decision processes (POMDPs), introduced in Sec. 2.5, are the primary paradigm for modeling and solving decision-making problems under uncertainty. POMDP-based decision-making can provide an optimal balance between exploration and exploitation, reducing uncertainty only when necessary to improve expected task performance. However, scientific domains often exhibit significant challenges, including:

(a) Extreme partial observability due to incomplete or noisy sensor observations,

(b) High-dimensional and continuous state spaces,

(c) Complex latent variable structure (highly coupled state and nuisance variables),

(d) Unpredictable, stochastic, or chaotic time-dynamics,

(e) Multiple spatial and temporal scales, and

(f) Complex human factors (scientific stakeholders, risk aversion, etc.).

Solving for optimal POMDP policies given a single one of these challenges would require innovation in state-of-the-art decision-making approaches. Unfortunately, real-world scientific data problems commonly exhibit all of these challenges simultaneously.

For example, consider again the example of mapping a deep sea hydrothermal plume with a marine robot carrying a chemical sensor (Fig. 1.2). The plume state is a complex, 3D structure (b) that is advecting due to currents on hourly scales and mixing due to turbulence on very short timescales (e). The plume expression in the water column changes moment by moment (d). The robot only observes the plume's state at a single point in space and time with its chemical sensor (a). Many observations will not contain plume water and the observations are highly correlated in space and time (c). Marine robot deployments require a significant investment of human and financial resources, and scientists and operators often require that robot behavior is interpretable and trustworthy (f). The robot must plan trajectories that collect informative point-observations of the plume, and fuse these observations together over time and space to generate a comprehensive model of the plume. This scientific decision-making problem is the focus of Chapter 3 in the thesis.

The Role of Uncertainty An implication of challenges (a-f) is that decision-makers in scientific domains must contend with significant environmental uncertainty, both aleatoric and epistemic. Choosing where to send the robot to take the next chemical observation is not as simple as examining the plume state and choosing a promising point. The planner fundamentally does not know where the plume is. The robot must instead explore the environment to reduce state uncertainty and then exploit that knowledge to choose promising observations. However, because of properties (a-f), exploration can be inefficient; many observations carry little information about the underlying, high-dimensional plume. Even as the robot collects observations and gathers information, the underlying plume is dynamic and changing; uncertainty will grow again. **Uncertainty is a fundamental and persistent part of scientific decision-making problems**.

The persistence of uncertainty in scientific decision-making makes many standard POMDP approximations infeasible. Due to this persistent uncertainty, the exploration-exploitation trade-off plays an outsized role in scientific decision making, compared to other traditional decision-making domains such as task and motion planning or manipulation. The decision-maker must reason about when and how to explore the environment to collect useful information that will effectively reduce state uncertainty and allow it to accomplish a given scientific task. It must also recognize when the current state estimate is sufficient should instead be exploited.

1.3 Task-Targeted Exploration

The unifying technical idea in this thesis is that not all uncertainty has equal impact in a decision-making problem; not all uncertainty is created equal. It may be critical to resolve specific aspects of state uncertainty and others may have no impact on the task or validity of the hypothesis. For example, if a scientist is trying to find the centerline of the plume — the region with the highest concentration of plume water — resolving the fine details of plume structure in the low concentrations regions far from the centerline is not necessary. If the scientific task, on the other hand, is to understand the extent of the plume spread, mapping the far-reaching, low-concentration regions would be critical. Given a task specified by loss or reward function, we develop decision-makers that can perform *task-targeted exploration*, allocating the minimal amount of exploration necessary to accomplish a task. This thesis presents novel algorithms for closed-loop scientific decision-making that perform parsimonious, targeted exploration in order maintain an acceptable level of task performance.

The ideas presented in this thesis enable task-targeted exploration and decision-making in real-world scientific domains under uncertainty. We develop three novel decision-making algorithms—in the fields of partially observable Markov decision processes and online learning—that balance exploration and exploitation efficiently in scientific domains, minimizing exploration while providing robust regret guarantees on algorithm performance.

These algorithms are principled, trustworthy decision-making paradigms that have been deployed in several real-world scientific domains, including deep sea hydrothermal plume discovery and climate and weather forecasting.

1.4 Contributions and Thesis Outline

This thesis develops decision-makers for autonomous scientific discovery that operate in highly uncertain environments and use task-targeted exploration to balance exploration and exploitation. Chapter 2 introduces background technical material on decision-making. In the following chapters, we analyze the exploration and exploitation trade-off in two different sequential decision-making formulations, online learning and partially observable Markov decision processes, and demonstrate that task-targeted exploration can lead to efficient and provably robust policies. The thesis presents the following contributions:

- **Chapter 3—PLUMES: Source-Seeking with Targeted Information Rewards** Hydrothermal plumes in the ocean are complex phenomena that experience advection by ocean currents, diffusion, and turbulent mixing. To effectively localize and map hydrothermal plumes, an autonomous underwater vehicle (AUV) must take exploratory sensing actions to localize the rising buoyant stem of the plume and exploit that knowledge to collect samples of the plume source. Chapter 3 formulates this "maximum-seek-and-sample" problem as a partially observable Markov decision process (POMDP) and presents PLUMES, an algorithm for nonmyopic planning of information gathering actions in a high-dimensional, continuous state space. PLUMES uses a maximum-value information reward to target exploration and drive the AUV to collect observations that efficiently identify the global maximum. Monte Carlo tree search with progressive widening is used for look-ahead planning in continuous observation spaces.

  In simulation and field deployments, we demonstrate that PLUMES successfully performs task-targeted exploration, allocating minimal exploration in order to identify the plume source and then exploiting that knowledge to collect samples at the source. Baseline algorithms, such as uniform coverage or upper confidence bound heuristics, are shown to over- or under-explore compared to PLUMES respectively. The developments in this chapter demonstrate that autonomous decision-making can advance oceanographic science. This work was originally presented in Flaspohler et al. [45].

- **Chapter 4—Value of Information and Macro-action Discovery in POMDPs**

Expanding upon the task-targeted exploration strategy developed in Chapter 3, this chapter introduces a novel *value of information (VoI)* metric that quantifies how a sensing or information gathering action affects long-term task success for general decision-making problems. VoI enables task-targeted exploration by explicitly approximating how sensitive task success is to exploratory actions.

We leverage VoI to construct high-level macro-actions[1] directly from a low-level POMDP model. Macro-actions are sequences of exploitative actions that can be leveraged whenever the VoI is low. We bound the performance of the resulting macro-action policies relative to an optimal policy. These macro-actions policies are applied to a simulated dynamic target tracking scientific application, in which a planner adapts its exploration strategy to the predictability of the underlying target. For highly unpredictable targets, the planner performs rapid, closed-loop sensing to keep state uncertainty low. For more predictable targets, the planner tolerates higher state uncertainty and only performs exploration infrequently. This work presents the first algorithm for macro-action discovery with performance guarantees in POMDPs and introduces a novel, quantitative connection between information and task performance via VoI. This work was originally presented in Flaspohler et al. [46].

- **Chapter 5—Online Learning with Optimism and Delay**

  The final chapter of the thesis deals with online learning. Online learning is a sequential decision-making paradigm in which a learner is pitted against an adversarial environment. POMDPs are a powerful model when the dynamic and observational models that underlie a domain are available and provide high-quality state estimates. However, in some domains, our ability to model future dynamics and observations is more limited. Adversarial online learning algorithms provide robust performance in many complex real-world online prediction problems such as climate or weather forecasting, where the predictability of future states of the world is inherently low due to chaos.

  The key to effective online learning is again to balance exploration and exploitation in the face of unknown and potentially adversarial future losses. The final methodological contribution of the thesis is developing robust online learning algorithms for real-world scientific applications. These algorithms—DORM, DORM+, and AdaHedgeD—automatically adjust their exploration-exploitation balance to target a given loss function, while dealing with the challenges of delayed feedback, short regret horizons,

[1] Macro-actions define sequences of exploitative actions that are executed in open-loop.

and real-time operational requirements. We demonstrate that these online learning methods produce zero-regret forecast ensembles for subseasonal climate forecasting and achieve robust, state-of-the-art performance from year to year. This work was originally presented in Flaspohler et al. [47]

Chapter 2

Background

This thesis deals with decision-making problems. The following chapter surveys classical and state-of-the-art algorithms for decision-making, spanning fields including online learning, bandit algorithms, sequential decision-making, and optimal experimental design. Although these problems are studied by different (and sometimes siloed) research communities, each with its own formulation and notation, there are deep connections between these branches of decision-making. We seek to make the connections between problems clear and present each paradigm within a unified notational and conceptual framework. This thesis makes contributions in the theory and practice of online learning (Chapter 5) and partially observable Markov decision processes (Chapters 3 and 4), but background material for several other decision-making paradigms are additionally included for completeness.

Human beings are autonomous agents that make sequences of choices to achieve their objectives. The field of decision-making or decision theory attempts to codify and formalize this process, such that it can be performed by computational systems. The study of computational decision-making has a long history and hundreds of papers and monographs have been published in past decades, e.g., [15, 18, 64, 174].

In the canonical decision-making formulation, an agent or decision-maker interacts with an environment with a potentially evolving state. The decision-maker selects actions (or plays or controls) and the environment then produces rewards and potentially additional observations (Fig. 2.1). This formulation is the basis of many problems in fields such as control theory, operations research, and reinforcement learning.

Decision-making problems can be classified by (a) how much information the decision-maker receives about the world, and (b) the properties of the decision-maker and environment (stationarity, persistence of state, controllability). In this chapter, we introduce four fundamental classes of decision-making problems:

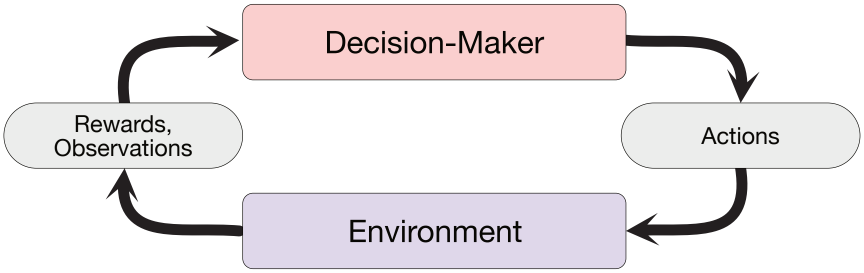


Figure 2.1: ***The Canonical Decision-Making Loop:*** *In the most general model of a decision-making process, a decision-maker or agent interacts with an environment by sequentially choosing actions; the environment in turn produces rewards and observations, which the decision-maker uses to improve its action selection.*

- **Online learning:** The decision-maker has no persistent state and the outcome of every action is fully observable at each timestep.
- **Bandit problems:** The decision-maker has no persistent state, but only the (potentially stochastic) outcome of a chosen action is observed.
- **Markov decision processes (MDPs):** The decision-maker has a persistent state that impacts available actions and rewards. The state of the environment and rewards are fully observable.
- **Partially observable Markov decision processes (POMDPs):** The decision-maker has a persistent state that impacts available actions and rewards. Rewards are fully observable but the state of the environment is only partially observable.

Other decision-making frameworks, such as partial monitoring [94] and reinforcement learning [174] are variants of these four fundamental classes with additional restrictions or capacities on the decision-maker and environment.

2.1 A Note on Related Work

The thesis covers a broad range of technical topics and scientific application areas. To maintain the readability and logical flow of the text, we have opted to place most of the related work locally within the thesis. Beyond what background and related work is presented in this chapter, the most relevant, contemporary work is highlighted in each technical chapter and the technical innovations of this thesis, as compared to past literature, are highlighted.

2.2 Online Learning

Online learning is a sequential decision-making paradigm in which a learner is pitted against a potentially adversarial environment [94, 129, 152]. At time t, the learner (decision-maker) must select an action, or play, $\boldsymbol{w}_t$ from some set of possible plays $\mathbb{W}$. The environment then reveals the loss function ℓ_t ("loss" is used conventionally in online learning, instead of "reward") and the learner pays the cost $\ell_t(\boldsymbol{w}_t)$. The learner uses information collected in previous rounds to improve its plays in subsequent rounds. Over a period of length T, the goal of the learner is to minimize *regret*, an objective that quantifies the performance gap between the learner and the best possible constant play in retrospect in some competitor set $\mathbb{U}$:

$$\mathrm{Regret}_T = \sup_{\boldsymbol{u} \in \mathbb{U}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}). \tag{2.1}$$

This adversarial formulation of online learning algorithms provides robust performance in many complex real-world online problems where predicting the behavior of the environment is challenging. Probabilistic formulations of online learning are also studied extensively [94, 129, 152]. In these formulations, distributional assumptions are made on the realized losses and the goal of the learner is to minimize expected regret, instead of worst-case regret (2.1).

Follow the regularized leader (FTRL) and online mirror descent (OMD) are two canonical algorithms for solving online learning problems [129, 152]. FTRL and OMD choose actions that achieve optimal worst-case regret against adversarial loss sequences by using the following decision rules to choose $\boldsymbol{w}_{t+1}$ based on the gradients of previously observed losses $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{w}_t)$, a strongly-convex regularizer ψ, and a regularization strength parameter λ:

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_{1:t}, \boldsymbol{w} \rangle + \lambda \psi(\boldsymbol{w}), \tag{FTRL}$$

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_t) \\ &\quad \text{with arbitrary} \quad \boldsymbol{w}_0, \end{aligned} \tag{OMD}$$

where $\mathcal{B}$ is the Bregman divergence: $\mathcal{B}_\psi(\boldsymbol{w}, \boldsymbol{u}) \triangleq \psi(\boldsymbol{w}) - \psi(\boldsymbol{u}) - \langle \nabla\psi(\boldsymbol{u}), \boldsymbol{w} - \boldsymbol{u} \rangle$.

The regularization strength parameter λ in these expressions plays the critical role of managing the trade-off between exploration and exploitation in online learning. For large values of the λ, the algorithms will favor values of $\boldsymbol{w}_t$ that minimize the regularizer and make more conservative plays (in some sense, exploring possible plays or "hedging their bets"). For small values of λ, the algorithms will favor values of $\boldsymbol{w}_t$ that perform well with respect to previously observed losses and make more risky plays (in some sense, exploiting their past knowledge). There are theoretical settings of λ that achieve optimal regret bounds, but in real-world applications these settings are often not practically useful and tuning the regularization strength parameter λ to achieve strong performance is challenging. State-of-the-art algorithms, such as AdaHedge [40, 129], attempt to tune λ online to improve the balance of exploration and exploitation as learning progresses.

Compared to the paradigms for sequential decision-making that appear later in this chapter, the online learning paradigm may look surprisingly sparse. There is no explicit model of an underlying environment or state, the system dynamics, or the observational model. However, this simplicity is what makes online learning formulations powerful. All information about the environment is distilled into a loss function, which encodes how well the chosen action accomplishes a given task. Instead of trying to model knowledge of current and future states of the environment, online learners select plays that guarantee small regret *under any adversarial realization of the underlying environment and resulting losses.* While this approach can be overly conservative when an accurate model of the underlying environment is available, online learning approaches are powerful in domains such as weather forecasting, where the underlying observational and dynamical models are imperfectly understood or chaotic.

2.3 Bandit Problems

Bandit problems are formulated identically to online learning problems, with one key difference: the environment only reveals the loss of the chosen action $l_t = \ell_t(\boldsymbol{w}_t)$ (*cf.* in online learning, the full loss function ℓ_t is revealed to the learner at each time step). Additionally, the loss of an action is often modeled as being stochastic, similar to stochastic online learning formulations. There is some underlying loss distribution associated with each play $\mathbb{P}_{\boldsymbol{w}_t}$, and the received loss is sampled from that distribution: $l_t \sim \mathbb{P}_{\boldsymbol{w}_t}$. These differences induce a different exploration-exploitation trade-off in bandit problems. The learner must strike a balance between playing a diverse set of actions in order to estimate their expected loss distributions, and exploiting that knowledge to make plays with low expected loss (or high expected reward).

As in online learning, the performance of bandit algorithms is measured using regret. In stochastic bandits, the chosen regret metric is often expected regret. We will switch to the more standard notation of bandit algorithms, in which the play is called an arm a and belongs to some set $\mathcal{A}$. The learner receives a reward r_t instead of a loss at each timestep. We will continue to use the notation $\mathbb{P}_a$ to represent the distribution over rewards under arm a.

Let $\mu^* = \max_{a \in \mathcal{A}} \mathbb{E}_{r \sim \mathbb{P}_a}[r]$ be the maximum expected reward of any arm $a \in \mathcal{A}$. Then the regret of a sequence of plays $(a_1, \ldots, a_T)$ on a bandit instance is:

$$\text{Regret}_T = \sum_{t=1}^{T} \mu^* - \mathbb{E}[R_t], \tag{2.2}$$

where the expectation is taken with respect to the probability distribution on rewards induced by the interaction between the bandit choices and the environment [94]. A bandit algorithm is said to be "no-regret" if the regret grows sublinearly, e.g., $\lim_{T \to \infty} \frac{1}{T}\text{Regret}_T = 0$. Beyond sublinearity, the specific growth rate of the regret for different algorithms has been studied extensively in bandit literature, and both upper- and lower-bounded [94].

Several bandit algorithms for the finite-arm, stochastic payoff domain achieve the lower bound on the rate of regret growth, including the popular upper confidence bound (UCB) algorithm [91, 94]. Other algorithms allow for adversarial formulations, as we saw in online learning, including the `Exp3` algorithm [8, 94]. Extensions to the basic finite stochastic setting allow for infinite arm sets, the inclusion of context, non-stationary payoff distributions, and a variety of other modifications. Lattimore and Szepesvári [94] provide an extensive overview of the theory and practice of bandit algorithms.

2.4 Markov Decision Processes

Both the online learner and the bandit algorithms played against environments that were "state-less", i.e., the choice made by the decision-maker at time t did not affect the available actions or future rewards in subsequent timesteps. While these online learning and multi-armed bandit formulations are very useful, many sequential decision-making problems arise in which an agent must consider the future consequences of actions taken on the problem state. In these formulations, the set of available actions and rewards are state-dependent and the state is controlled by the decision-making process. For example, a marine robot that drives into a long channel searching for a target has to contend with the consequence of backtracking and re-localizing if the target is not found in the channel. Sequential decision-making problems with state are ubiquitous in real-world applications, from robotics to agriculture to personalized medicine.

Sequential decision-making with state has been studied for nearly a century, starting with the seminal works of Bellman [15], Howard [64] in the early 1950s (although the field has roots back to the 17th century [143, Ch. 1]). A variety of formulations have been developed for specifying sequential decision-making problems. Perhaps the most pervasive are "classical planning" problems—often encoded using a programming domain definition language (PDDL), Markov decision processes (MDPs), and partially observable Markov decision processes (POMDPs). Each formulation admits increasing levels of problem uncertainty: PDDL solvers often assume that state transitions deterministically given a chosen action; MDPs can model stochastic state transitions; and POMDPs allow the state itself to be partially observed via noisy and limited sensors by introducing a stochastic measurement model. Many scientific applications have state which is best modeled as stochastic and partially observable; therefore, Chapters 3 and 4 focus on the full POMDP problem. The MDP formulation and its extension to POMDPs are introduced in the following sections.

The key idea in Markov decision process is that the environment has a state that impacts the rewards received when different actions are taken and impacts future states. MDPs are *Markov* in the sense that future states depend on past states only via the current state, i.e., letting S_t represent the state at time t and given an action a, then $\mathbb{P}(S_{t+1} \mid S_0 = s_0, \dots, S_t = s_t, a) = \mathbb{P}(S_{t+1} \mid S_t = s_t, a)$. This Markov property of the state is key for developing efficient solvers for MDP problems.

Formally, an MDP is a tuple: $(\mathcal{S}, \mathcal{A}, T, R, H, \gamma)$:

- $\mathcal{S}$: a set of discrete or continuous states,
- $\mathcal{A}$: a set of discrete or continuous actions,
- T: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, the transition function, i.e., $T(s, a, s') = \mathbb{P}(S_{t+1} = s' \mid S_t = s, a_t = a)$,
- R: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the reward of taking action a when the state is s, i.e., $R(s, a)$,

- H: the horizon, $H \in [0, \infty]$, and
- γ: discount factor, $0 \leq \gamma \leq 1$; for infinite horizon problems we often require $\gamma < 1$.

In the remainder of the text, we present the finite-horizon MDP formulation. Infinite horizon MDPs are an extension of the finite horizon case, given basic regularity assumptions (i.e., the discount factor ensures that the infinite expected reward converges, or the agent is guaranteed to enter an absorbing state with zero reward).

As in many formulations of decision making, the MDP objective is to maximize the expected reward over the planning horizon, given the distribution over initial states. Solving an MDP involves producing a (potentially horizon-dependent) policy $\{\pi_t : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}_{t=0}^{H-1}$, mapping from states to a distribution over actions. MDP policies are generally evaluated by their expected, discounted, cumulative reward, given initial state s. This objective is known as the MDP *horizon-h value function under policy* π, starting from state s:

$$V_h^\pi(s) = \mathbb{E}\Big[\sum_{t=0}^{h-1} \gamma^t R\big(s_t, \pi_t(s_t)\big) \mid s_0 = s\Big], \tag{2.3}$$

where the expectation is taken over the random outcomes of interactions between the environment (the MDP model) and the policy.

Evaluating and optimizing the value function, as formulated in (2.3), is seemingly complex, as it requires computing the distribution over outcomes that arise from a policy interacting with the MDP environment while simultaneously optimizing that policy. A key innovation in the theory of MDPs was the recognition that (2.3) could be re-written recursively:

$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \int_{s' \in \mathcal{S}} V_{h-1}^\pi(s') \; \mathbb{P}(s' \mid s, \pi(s)) \; \mathrm{d}s'. \tag{2.4}$$

The recursive expression (2.4) is the *Bellman equation* for the MDP value function. It decomposes the long-term value of a state into a sum of its instantaneous reward and the value of the subsequent states at horizon $h - 1$, weighted by the probability of reaching those states given the current state and chosen action.

The value function induces a partial ordering on the set of possible policies, i.e., a policy π is better or equal to a policy π' if the value of π is greater than or equal to π' for all states [174], and an optimal policy maximizes the value function over the set of possible actions for all states. There is a large body of literature that studies the existence and uniqueness of optimal MDP policies [17, 94, 135, 143]. Although general MDP policies may be stochastic, for a large class of real-world MDPs, there exists a deterministic, memoryless optimal policy [94]. In the remainder of the thesis, we will focus on the class of deterministic, memoryless policies.

The optimal MDP value function is given by maximizing (2.4) over the set of possible actions:

$$V_h^*(s) = \max_{a \in \mathcal{A}} R(s,a) + \gamma \int_{s' \in \mathcal{S}} V_{h-1}^*(s') \, \mathbb{P}(s' \mid s,a) \, \mathrm{d}s', \tag{2.5}$$

and an optimal MDP policy is given by choosing actions that are greedy according to a one-step look ahead of the optimal value function: $\pi^*(s) = \arg\max_{a \in \mathcal{A}} R(s,a) + \gamma \int_{s' \in \mathcal{S}} V_{h-1}^*(s') \, \mathbb{P}(s' \mid s,a) \, \mathrm{d}s'$. Algorithms such as policy iteration and value iteration ([174]) can be used to approximate the optimal MDP value function and policy.

MDPs are a highly flexible formulation of a decision-making that can represent a diverse set of real-world problems. The simplest version of an MDP has discrete states and actions. This is known as a “tabular MDP”, because the transition function can be represented by a set of transition matrices $\{\mathbf{T}_a\}_{a=0}^{|\mathcal{A}|-1}$, such that $\mathbf{T}_a[i,j] = \mathbb{P}(S_{t+1} = i \mid S_t = j, a_t = a)$ and the reward function by a reward matrix $\mathbf{R}$, such that $\mathbf{R}[i,j] = R(s_i, a_j)$. When the size of the state and action spaces are small enough to fit these objects in computer memory, tabular MDPs can be solved efficiently using value or policy iteration.

2.5 Partially Observable Markov Decision Processes

When the state is not fully observable in a sequential decision-making problem, the standard MDP formulation is no longer applicable. Partially observable states arise in many practical decision-making problems in which the environment is observed through limited or noisy sensors. In these problems, the state is no longer available as a sufficient statistic for decision-making. Instead, the decision-maker receives observations that reveal some partial information about the underlying state. The decision-maker must make use of its *history* of actions and observations at time t, $H_t = (a_0, z_1, a_1, \ldots, z_{t-1}, a_{t-1}, z_t)$ to make subsequent decisions.

The key difficulty in so-called partially observable Markov decision processes (POMDPs) is that the optimal decision at time t depends on the full history of observations and actions taken by the agent. Learning history-dependent policies, instead of state-dependent polices, is computationally challenging: the number of possible histories grows exponentially in the planning horizon. This is known as the “curse of history” in the POMDP literature.

Many POMDP models introduce a *belief state*, which is a probabilistic representation of the current interaction history. The belief is a sufficient-statistic for decision-making [7] and, when the belief is used to represent the state of the world, the Markov property of the state is restored. However, beliefs are represented by (often high-dimensional) probability distributions. Planning, therefore, must happen in high-dimensional, continuous belief space. In discrete problems, the size of the belief space grows exponentially with the number of discrete states. This is known as the "curse of dimensionality" in the POMDP literature. These twin curses—of history and dimensionality—suffice to make decision-making in partially observable environments challenging.

Formally, a finite horizon POMDP can be represented as tuple: $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O, b_0, H, \gamma)$, where $\mathcal{S}$ are the states, $\mathcal{A}$ are the actions, R is the reward, H is the horizon, and γ is the discount factor, as defined in Sec. 2.4. Additionally, we define:

- $\mathcal{Z}$: a set of discrete or continuous observations,
- O: $\mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$, the observation or measurement function, i.e., $O(s, a, z) = \mathbb{P}(Z_t = z \mid S_t = s, a_{t-1} = a)$,
- b_0: the initial belief state, i.e., $b_0 \in \mathcal{P}(\mathcal{S})$.

As the agent moves through the world, it selects actions and receives observations. Because the state of the world is not directly observable, the agent maintains a *belief* and must update this belief each time it takes an action and receives an observation. Given the transition and observation models, the belief can be updated directly using Bayes rule (the belief is the posterior distribution on the current state, conditioned on the history). We define the belief update function as follows:

$$\begin{aligned}
b_t(s) &= \tau(b_{t-1}, a_{t-1}, z_t)(s) && (2.6)\\
&\triangleq \mathbb{P}(S_t = s \mid a_0, z_1, \ldots, z_{t-1}, a_{t-1}, z_t) && (2.7)\\
&= \mathbb{P}(S_t = s \mid b_{t-1}, a_{t-1}, z_t) && (2.8)\\
&= \frac{\int_{s' \in \mathcal{S}} O(s, a_{t-1}, z_t) T(s', a_{t-1}, s) b_{t-1}(s') \,\mathrm{d}s'}{\mathbb{P}(z_t \mid b_{t-1}, a_{t-1})} && (2.9)
\end{aligned}$$

This equation is known as a Bayes filter [71] and is often intractable to compute directly in large or continuous state spaces, as the data evidence term in the denominator is expensive to compute. An approximate Bayesian inference procedure or filter can be used to represent the belief, such as a Kalman filter [79], particle filter [83], or more general MCMC or variational methods [21, 56, 73].

As in the MDP formulation, the realized reward in a POMDP is a random variable. Optimal planning is defined as finding a belief-dependent policy $\{\pi_t^* : \mathcal{P}(\mathcal{S}) \to \mathcal{A}\}_{t=0}^{H-1}$ that maximizes expected reward: $\mathbb{E}\Big[\sum_{t=0}^{H-1} \gamma^t R\big(S_t, \pi_t(b_t)\big) \mid b_0\Big]$, where b_t is the updated belief at time t, conditioned on the history of actions and observations. The recursively defined horizon-h optimal POMDP value function V_h^* quantifies, for any belief b, the expected cumulative reward of following an optimal policy over the remaining planning iterations: $V_0^*(b) = \max_{a\in\mathcal{A}} \mathbb{E}_{s\sim b}[R(s,a)]$ and

$$V_h^*(b) = \max_{a\in\mathcal{A}} \mathbb{E}_{s\sim b}[R(s,a)] + \gamma \int_{z\in\mathcal{Z}} \mathbb{P}(z \mid b, a) V_{h-1}^*(\tau(b,a,z))\,\mathrm{d}z \qquad h \in [1, H-1], \quad (2.10)$$

where $\tau(b,a,z)$ is the updated belief after taking action a and receiving observation z (2.9). Again, as in the MDP formulation, the optimal policy at horizon h is to act greedily according to a one-step look ahead of the horizon-h value function.

2.5.1 Piecewise-Linear and Convex Value Functions

Despite its continuous nature, the value function for any discrete, finite horizon POMDP can be represented by a piecewise-linear and convex function (PWLC) with a finite number of supporting hyperplanes, often called α-vectors [77, 160]. Value iteration over belief space proceeds by building the horizon-h value function from the set of α-vectors at horizon-$(h-1)$. At each step of value iteration, the resulting value function remains PWLC [77].

This property has been leveraged to compute optimal policies in small, discrete POMDPs and can be proved by induction. As for tabular MDPs, we represent the discrete transition function T by a set of transition matrices $\{\mathbf{T}_a\}_{a=0}^{|\mathcal{A}|-1}$, such that $\mathbf{T}_a[i,j] = \mathbb{P}(S_{t+1} = i \mid S_t = j, a_t = a)$. The observation function O will be represented by a set of observation matrices $\{\mathbf{O}_a\}_{a=0}^{|\mathcal{A}|-1}$, such that $\mathbf{O}_a[i,j] = \mathbb{P}(Z_t = i \mid S_t = j, a_{t-1} = a)$.

Base Case. For each action a, construct an α-vector: $\alpha_a = [R(s_1,a), \ldots, R(s_N,a)]^\top$ for each state $s_1, \ldots, s_N \in \mathcal{S}$. Then, by definition of the vector inner product:

$$V_0^*(b) = \max_{a\in\mathcal{A}} \sum_{s\in\mathcal{S}} b(s) R(s,a) = \max_{a\in\mathcal{A}} \alpha_a^\top b. \quad (2.11)$$

As the maximum of $|\mathcal{A}|$ linear segments, V_0^* is PWLC and is represented by the supporting hyperplanes $\Gamma_0 = \{\alpha_a\}_{a=0}^{|\mathcal{A}|}$. The size of the α-vector set is $|\Gamma_0| = |\mathcal{A}|$.

Inductive Case. Assuming that V_{h-1}^* is PWLC and can be represented by a finite set of α-vectors Γ_{h-1}, we show that V_h^* is PWLC. Recall from (2.10):

$$V_h^*(b) = \max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} b(s) R(s,a) + \gamma \sum_{z \in \mathcal{Z}} \mathbb{P}(z \mid b, a) V_{h-1}^*(\tau(b,a,z)) \qquad h \in [1, H-1]. \tag{2.12}$$

We can re-write (2.12) using the transition and observation matrices and the α-vectors defined in (2.11):

$$\begin{aligned} V_h^*(b) &= \max_{a \in \mathcal{A}} \alpha_a^\top b + \gamma \sum_{z \in \mathcal{Z}} \mathbb{P}(z \mid b, a) V_{h-1}^* \Big(\frac{\operatorname{diag}(\mathbf{O}_a[z,:]) \mathbf{T}_a b}{\mathbb{P}(z \mid b, a)} \Big) && (2.13) \\ &= \max_{a \in \mathcal{A}} \alpha_a^\top b + \gamma \sum_{z \in \mathcal{Z}} \mathbb{P}(z \mid b, a) \max_{\alpha \in \Gamma_{h-1}} \alpha^\top \Big(\frac{\operatorname{diag}(\mathbf{O}_a[z,:]) \mathbf{T}_a b}{\mathbb{P}(z \mid b, a)} \Big) && (2.14) \\ &= \max_{a \in \mathcal{A}} \alpha_a^\top b + \gamma \sum_{z \in \mathcal{Z}} \cancel{\mathbb{P}(z \mid b, a)} \cancel{\frac{1}{\mathbb{P}(z \mid b, a)}} \max_{\alpha \in \Gamma_{h-1}} \alpha^\top \operatorname{diag}(\mathbf{O}_a[z,:]) \mathbf{T}_a b && (2.15) \\ &= \max_{a \in \mathcal{A}} \alpha_a^\top b + \gamma \sum_{z \in \mathcal{Z}} \max_{\alpha \in \Gamma_{h-1}} \alpha^\top \operatorname{diag}(\mathbf{O}_a[z,:]) \mathbf{T}_a b. && (2.16) \end{aligned}$$

where (2.13) is due to the definition of the belief transition operator (2.9), (2.14) makes use of the inductive hypothesis that V_{h-1}^* is PWLC and can be represented with a finite set of α-vectors Γ_{h-1}^*, and (2.15) follows because the normalizing data evidence factor does not affect the maximum and can be pulled out to cancel with the term from the expectation.

The second part of expression (2.16) yields a unique α-vector for each distinct value of $a \in \mathcal{A}$, $z \in \mathcal{Z}$, and $\alpha_i \in \Gamma_{h-1}$:

$$\alpha_{a,z,i}^\top = \gamma \alpha_i^\top \operatorname{diag}(\mathbf{O}_a[z,:]) \mathbf{T}_a. \tag{2.17}$$

Any of these vectors could be maximizing *a priori* for a specific belief b. Therefore, the α-vector set at horizon h must contain the Cartesian sum of all possible maximizing α_i's for each $z \in \mathcal{Z}$:

$$\Gamma_a = \Big\{ \alpha_a + \alpha \mid \alpha \in \alpha_{a,z_1,\cdot} \oplus \alpha_{a,z_2,\cdot} \oplus \cdots \oplus \alpha_{a,z_M,\cdot} \Big\} \tag{2.18}$$

The final set of α vectors for horizon h is the union over action choices:

$$\Gamma_h = \bigcup_{a \in \mathcal{A}} \Gamma_a. \tag{2.19}$$

The set of unique vectors in (2.17) is of size $|\mathcal{A}||\mathcal{Z}||\Gamma_{h-1}|$ and (2.18) produces $|\mathcal{A}||\mathcal{Z}|^{|\Gamma_{h-1}|}$ unique vectors. Therefore, the set Γ_h has size $|\Gamma_h| = \mathcal{O}(|\mathcal{A}||\mathcal{Z}|^{|\Gamma_{h-1}|})$ and grows exponentially in the number of discrete actions $\mathcal{A}$ and doubly exponentially in the number of discrete observations $\mathcal{Z}$ with the planning horizon H. ■

2.5.2 Approximate Solvers

As we have seen, (2.10) is intractable for POMDPs with continuous state, action, or observation spaces; for large discrete problems, the doubly exponential growth in the number of α-vectors possibly needed to represent the optimal value function (Sec. 2.5.1) makes optimal policy computation infeasible [77, 135]. Thus, an optimal policy for POMDPs must be approximated. Much of the art of practical decision-making under uncertainty is making well-designed algorithmic and heuristic choices that enable efficient and robust planning algorithms. These algorithms can be broken into offline and online solvers and are summarized in the following sections and in Table 2.1.

Offline Solvers

POMDP solvers can be grouped into two categories: offline and online solvers. Offline solvers construct a policy entirely offline, before the agent begins interacting with the environment. Because the solver must commit to a policy without seeing what belief states the agent realizes, offline solvers have to produce policies that perform well from any given belief state. This inability to tailor the policy approximation to belief states that are realized during execution is the main drawback of offline POMDP solvers; the produced policy must be globally good. The advantage of offline solvers is that they generally have access to significant computational resources and are free from size, weight, and power (SWaP) constraints that online solvers often face if they are deployed on robotic platforms or in real-world scenarios.

The most common offline POMDP solvers perform some form of value iteration using a piecewise-linear and convex approximation of the value function made up of α-vectors. These algorithms include the exact solvers of Kaelbling et al. [77], Sondik [160] and a host of approximate solvers, detailed in Table 2.1. Each approximate solver tries to exploit some structure in the POMDP value function to produce policies that approximate the optimal policy as closely as possible. Although the development of approximate, offline POMDP solvers has progressed in the last several decades, state-of-the-art algorithms still suffer from significant scalability challenges and have seen limited application to real-world decision-making problems.

Online Solvers

Unlike offline solvers, online POMDP solvers construct an approximate policy online: given the current belief state b_t, an online solver will seek to approximate the optimal policy only at the current belief $\pi^*(b_t)$. In this way, online solvers can (at least partially) circumvent the curse of dimensionality by focusing computation on the region of belief space that is locally reachable from b_t. Additionally, online solvers often search over a limited horizon $h < H$. This limited look-ahead search helps to circumvent the curse of history. The penalty paid by online solvers for these benefits is that the policy search for $\pi^*(b_t)$ must be done at time t, while the agent is interacting with the world. In some applications, this penalty may not be steep; a resource manager may make a choice about acquisition only once a day or month, and the planner has sufficient time to develop a new approximate policy before a decision must be made. On the other hand, if the decision-maker is driving a robot through the world, the solver may only have seconds to collect an observation, update its belief, and produce a new policy. When resource constraints are significant, online POMDP planning may be infeasible or require only very limited local search with short, look-ahead horizons. In these cases, hybrid offline-online planners that can make use of additional computational resources in an offline setting may be more practical.

The majority of modern, online POMDP solvers are based on tree search, inspired by the success of algorithms like Monte Carlo tree search [25]. These algorithms approximate the value function at a given belief by forward simulating different action choices and resultant observations. Different algorithms leverage different heuristics to chose how to forward simulate experiences and how to estimate the POMDP value function based on sampled experiences. A summary of several influential online POMDP algorithms is given in Table 2.1.

Table 2.1: ***POMDP Algorithms:*** *Selected POMDP planning methods, organized by algorithm name, whether the algorithm operates offline or online, whether the algorithm can support continuous states ($\mathcal{S}$), actions ($\mathcal{A}$), and observations ($\mathcal{Z}$), and a summary of the key idea enabling the algorithm.*

Alg.	Off/online	Cont. $\mathcal{S}$, $\mathcal{A}$, $\mathcal{Z}$?	Key Idea
PBVI [137]	Offline	N,N,N	Maintain α-vectors that dominate for a specific set of exemplar beliefs. Choose these beliefs to approximate the reachable belief space.
SARSOP [90]	Offline	N,N,N	Maintain α-vectors that dominate for a specific set of exemplar beliefs. Maintain upper and lower bounds on the value function and sample these beliefs to approximate the optimally reachable belief space.
MCVI [12]	Offline	Y,N,Y	Replace value function policy representation with policy graph. Use α-vectors to represent value. Approximate α-vectors via Monte Carlo (MC) rollouts. Can handle large spaces due to MC sampling.
Perseus [140, 162]	Offline	Y,Y,Y	Instead of keeping an α-vector for every belief, maintain only some subset of vectors necessary to increase the value estimate at every iteration. To extend to continuous spaces, leverage Gaussian mixture representations, sampled actions, and discretized observations.
Hilbert POMDPs [127]	Offline	Y,Y,Y	Use kernel embedding to represent belief and value function.
MC POMDPs [24]	Offline	Y,Y,Y	Use MC sampling to estimate conditional values and importance sampling (IS) to correct for observation sampling bias. Learn a high-level discrete state decomposition based on task.
DESPOT [159]	Online	Y,N,N	Construct a sparse policy tree using sampled scenarios. Estimate value by constructing tree and doing backups. Do state rollouts instead of belief rollouts.
DESPOT-α [49]	Online	Y,N,Y	Extend DESPOT to larger observation spaces the using α-vectors to represent value at intermediate tree nodes.
POMCP [155]	Online	Y,N,N	Construct a sparse policy tree using Monte Carlo sampling and upper confidence tree heuristics at intermediate nodes. Estimate value by doing rollouts and backups. Do state rollouts instead of belief rollouts.
POMCPOW [172]	Online	Y,Y,Y	Similar to POMCP but use double progressive widening to handle continuous observation and actions.

2.5.3 Open versus Closed-loop Planning

In their traditional formulation, both MDPs and POMDPs model “closed-loop” decision-makers—decision-makers that observe the state or collect an observation at each timestep and are able to update their belief and policy based on the received feedback. During planning, a closed-loop decision-maker will anticipate receiving information about the state, either through direct observation or a partially observable observation model, and can plan information-gathering actions in order to improve task performance.

On the other extreme, “open-loop” decision-makers commit to an action sequence or plan and execute that plan without receiving state feedback or observations. Open-loop planners can either ignore the uncertainty about future state in MDP and POMDP formulations (e.g., planning for the maximum-likelihood estimate of the state evolution), or by anticipating the lack of observation, generate plans that are good in expectation over all possible future state realizations. Open-loop planners produce a plan or action sequence; closed-loop planners produce a policy.

There are solutions that sit between these two extremes, known as open-loop feedback planning (or model-predictive control in the controls community). Open-loop feedback planners generate open-loop plans over a receding horizon, but collect state information or observations and re-plan at some regular interval. An open-loop feedback planner still generates a plan, instead of a policy, but is able to make use of state information or observation when it is available. When full, closed-loop planning is intractable but open-loop planning is possible, open-loop feedback planners are often a practical compromise.

2.6 Experimental Design

Experimental design is the problem of selecting a set of measurements or experiments in order to learn about an unknown system. Experimental design has for centuries been performed largely by human scientists, who leverage domain knowledge, intuition, and codified best-practice to design informative experiments. Starting in the 1960s, the experimental design problem was abstracted and formalized as an optimization problem, given a model of the unknown system and measurement process. The formalization of experimental design as a decision-making problem is known as optimal experimental design (OED), or Bayesian optimal experimental design (BOED) [27, 142].

OED deals with models of the form:

$$y = f(\boldsymbol{x}, \theta), \tag{2.20}$$

where $y \in \mathcal{Y} \subseteq \mathbb{R}$ is a scalar response variable (or observation), $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^m$ is the specification of the experiment conditions (or actions), and $\theta \in \Theta \subseteq \mathbb{R}^d$ are unknown parameters that determine the state of the environment but are not controllable by the experimenter. The observation function $f : \mathcal{X} \times \Theta \to \mathcal{Y}$ may generally be complex and nonlinear.

In the special case that f is restricted to have a linear dependence on θ and the observation y is corrupted by statistical noise, we arrive at a class of OED problems known as *statistical linear experimental design*:

$$y = g(\boldsymbol{x})^\top \theta + \epsilon, \tag{2.21}$$

for arbitrary function $g : \mathcal{X} \to \mathbb{R}^d$ and noise distribution $\epsilon \sim \mathbb{P}_\epsilon$. The noise distribution $\mathbb{P}_\epsilon$ is often assumed to have zero mean, finite variance, and a distribution that is independent of the chosen experimental conditions $\boldsymbol{x}$.

If we run n experiments with different experimental conditions $(g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n))$, we can model the full experimental set up with an $n \times d$ *design matrix* $\mathbf{X}$, a $n \times 1$ *response vector* $\boldsymbol{y}$, an $n \times 1$ *error vector* $\boldsymbol{\epsilon}$ as:

$$\boldsymbol{y} = \mathbf{X}\theta + \boldsymbol{\epsilon}. \tag{2.22}$$

If we further assume that the observation noise has a Gaussian distribution, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, the problem is known as *normal linear experimental design.* This is the most commonly studied class of OED problems, as several elegant and computationally inexpensive algorithms exist for selecting the design matrix $\mathbf{X}$.

For the normal linear problem specified in (2.22), the minimum variance unbiased (MVU) estimator of θ given a design matrix and a response vector is:

$$\hat{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{y}. \tag{2.23}$$

This estimator has error covariance:

$$\mathbb{E}[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^\top] = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1} = \sigma^2 \Big(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \Big)^{-1}. \tag{2.24}$$

The basic method behind normal linear OED is to define a scalarization of the error covariance matrix and optimize that scalarization with respect to the design matrix $\mathbf{X}$. Different scalarizations focus on different measures of the "size" of the error covariance matrix:

- **A-optimality:** minimize the trace of the error covariance
- **D-optimality:** minimize the determinant of the error covariance
- **E-optimality:** minimize the norm of the error covariance, i.e., the maximum eigenvalue

Given a scalarization choice, the design matrix is then optimized using a variety of techniques, including greedy selection and convex optimization as each of A-, D-, and E-optimality give rise to a convex objective function over a convex set [23, Section 7.5].

In contrast to the frequentist OED formulation, Bayesian OED [27] places a prior distribution on the environment, models the probabilistic measurement model for the unknown parameters θ, and seeks to optimize an expected utility function, where the expectation is taken with respect to the posterior distribution on θ given the chosen observations. For environmental models including Gaussian processes, Gaussian Markov random fields, and other Bayesian models, past approaches have used submodular optimization and greedy selection to maximize information criteria and solve BOED problems [53, 87, 88, 132, 134, 192, 193, 199, 200],

2.6.1 Connections to other fields of decision-making

The problem of experimental design has deep connections to other decision-making paradigms that we have presented in this section. We can classify standard experimental design as being "stateless", i.e., the experiment selected at time t does not impact the available experiments in future times or the state of the environment. However, the "reward" in an experimental design problem is not additive: if the same experiment is selected multiple times, the information provided about the unknown parameters diminishes. This complicates decision-making. On the other hand, in the normal linear setting, the "reward" from each play is a known, non-random quantity. This makes normal linear OED amenable to batch optimization-based solutions, unlike bandit problems in which the reward is randomly sampled from an unknown payoff distribution [94, Section 4.2].

In non-normal, non-linear experimental design problems, the reward of each design $\boldsymbol{x}$ may depend on the random observation received y. These formulations are more like a standard bandit problem and are often solved using sequential approaches, such as greedy selection. Other extensions of the basic OED problem may include a state if the experiments directly impact the state of the environment or the ability to run future experiments. For these problems, full POMDP-style planning becomes necessary. However, again, the reward in OED problems often encapsulates the information gathered about the unknown parameters and is not additive. Extensions of the standard, additive, state-based reward POMDP model presented in Sec. 2.5 to information rewards or other non-additive, belief-based rewards are given in Araya et al. [4], Fehr et al. [43]. These formulations may be appropriate for non-linear, non-normal experimental design problems with state.

Chapter 3

PLUMES: Source-Seeking with Targeted Information Rewards

In many environmental and earth science applications, experts want to collect scientifically valuable samples to characterize a source or concentration maximum of an initially unobserved environmental phenomena. We will call this problem of seeking and characterizing a source phenomena the *maximum seek-and-sample* (MSS) problem. The MSS problem arises in applications from hydrothermal vent discovery to oil spill response to volcanic monitoring (Fig. 3.1).

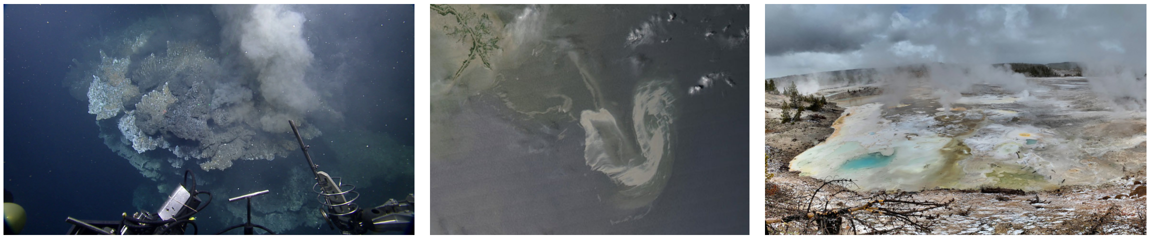


Figure 3.1: ***Source Seeking Applications:*** *Deep sea hydrothermal plumes (left), oceanographic oil-spills (center), and volcanic fumarole basins (right) each give rise to multimodal expressions from single or multiple potentially time-varying sources. Image credit: (Left) WHOI JASON Team; (Center) NASA/GSFC/MODIS Rapid Response Team; (Right) WHOI, Adam Soule and William Pardis.*

Robustly and generally solving the MSS problem is challenging because it is fundamentally a problem of balancing exploration and exploitation in an unknown environment. Many phenomena, including the three highlighted in Fig. 3.1, can only be observed using single-point chemical sensors. Using these limited sensors to find the maximum concentration point in a complex, multimodal environment requires solving the dual problems of (1) mapping the unknown environment, and (2) using that map to localize and characterize a potentially time-varying source. However, a key challenge and opportunity in the MSS problems is that a perfect map of the environment is not necessary to robustly localize a source. Therefore, a scientist or autonomous decision-maker must answer the question: *how well do I need to map this environment to ensure that I have localized the global maximum point or source?* Seeking an effective balance between exploration and exploitation in the MSS problem requires task-targeted exploration. The development of task-targeted exploration strategies in this specific application will inspire the general task-targeted exploration approach that we develop in Chapter 4.

In canonical MSS solutions, the exploration-exploitation trade-off is resolved by choosing to favor exploration. Observations are collected at predetermined locations by a technician or mobile platform following a uniform coverage trajectory. These non-adaptive strategies over-explore, resulting in sample sparsity at the source. Additionally, executing a fixed, pre-planned trajectory may be infeasible when the geometric structure of the environment is unknown (e.g., boulder fields) or changing (e.g., tidal zones). Increasing the number of valuable samples that localize and characterize a source requires adaptive online planning that strikes a more refined balance between exploration and exploitation.

To enable adaptive source-seeking in partially observable environments, this chapter formulates the MSS problem as a partially observable Markov decision process (POMDP). POMDPs are general models for decision-making under uncertainty, which we have reviewed in Sec. 2.5. We define the MSS POMDP, in which the partially observable state represents the continuous environmental phenomenon and a sparse reward function encodes the MSS scientific objective by giving reward only to samples sufficiently close to the global maximum. Solving for an optimal POMDP policy is generally intractable [124], and the MSS POMDP is additionally complicated by both continuous state and observation spaces, the sparse MSS reward function, and limited and noisy sensor observations.

To overcome these challenges, this thesis presents PLUMES — **P**lume **L**ocalization under **U**ncertainty using **M**aximum-valu**E** information and **S**earch — an adaptive algorithm that approximately solves the MSS POMDP and enables a mobile robot to efficiently localize and densely sample an environmental maximum, subject to practical challenges including dynamic constraints, unknown geometric map and obstacles, and noisy sensors with limited field-of-view. The key to PLUMES is the use of a targeted information reward that drives a robot or autonomous sensor to perform the minimum amount of exploration necessary to confidently localize the maximum point in an continuous, multi-model environment.

Motivating Application: Plume Stem Localization. Hydrothermal vents in the ocean were first observed in 1977 [33] at the Galapagos Rift, and since have been a concerted focus of a geodynamical and biogeochemical studies. Hundreds of undiscovered vent sites are hypothesized to exist in the deep sea [14] with implications for global nutrient and energy budgets, and novel ecosystems. The problem of vent localization has been addressed by many authors (e.g., [69, 82, 112, 113, 125, 133, 185]) and is a challenging problem given the complicated spatiotemporal plume structure caused by tidal advection, water mass mixing, and chaotic turbulence in hydrothermal plume environments.

Hydrothermal plumes in the deep sea are typically characterized as buoyancy-driven water masses. At a vent source, emitted fluids are significantly less dense than background seawater (primarily by virtue of being super-heated). This less dense water mass rises rapidly in the water column, forming a *buoyant stem* (Fig. 3.2, A), which grows approximately 1 m in diameter for every 10 m vertically travelled. Due to rapid cooling, turbulent mixing, and the natural stratification of ocean water, vent-derived waters will reach a point of neutral-buoyancy with the background seawater. At this point, the plume forms a *nonbuoyant or neutrally buoyant or intrusion layer* (Fig. 3.2, A & B), which spreads out across the isopycnal that describes the ocean layer of equivalent density. In the Atlantic basin, plume rise height is typically expected to be approximately 300-350 m; in the Pacific basin, this is 150-200 m [163]. This buoyant stem and neutrally-buoyant layer model of a hydrothermal plume has been mathematically codified perhaps most famously by Morton et al. [122] as a system of conservative equations.

In most real-world environments, advective cross-flow is present. This "bends" a buoyant stem in a time-varying way and reduces the effective rise height of the plume by introducing more aggressive mixing [179]. A time-varying cross-flow will move the location of the buoyant stem within the large, intrusion layer and leads to complicated, multimodal structure within the intrusion layer (Fig. 3.2, D). We will model the problem of buoyant stem localization within a multimodal intrusion layer as a MSS POMDP.

Overview. The remainder of this chapter presents PLUMES, a planner for localizing and collecting samples at the global maximum of an *a priori* unknown and partially observable continuous environment. We formulate the MSS problem as a partially observable Markov decision process (POMDP) with continuous state and observation spaces, and a sparse reward signal. To solve the MSS POMDP, PLUMES uses an information-theoretic reward heuristic with continuous-observation Monte Carlo Tree Search to efficiently localize and sample

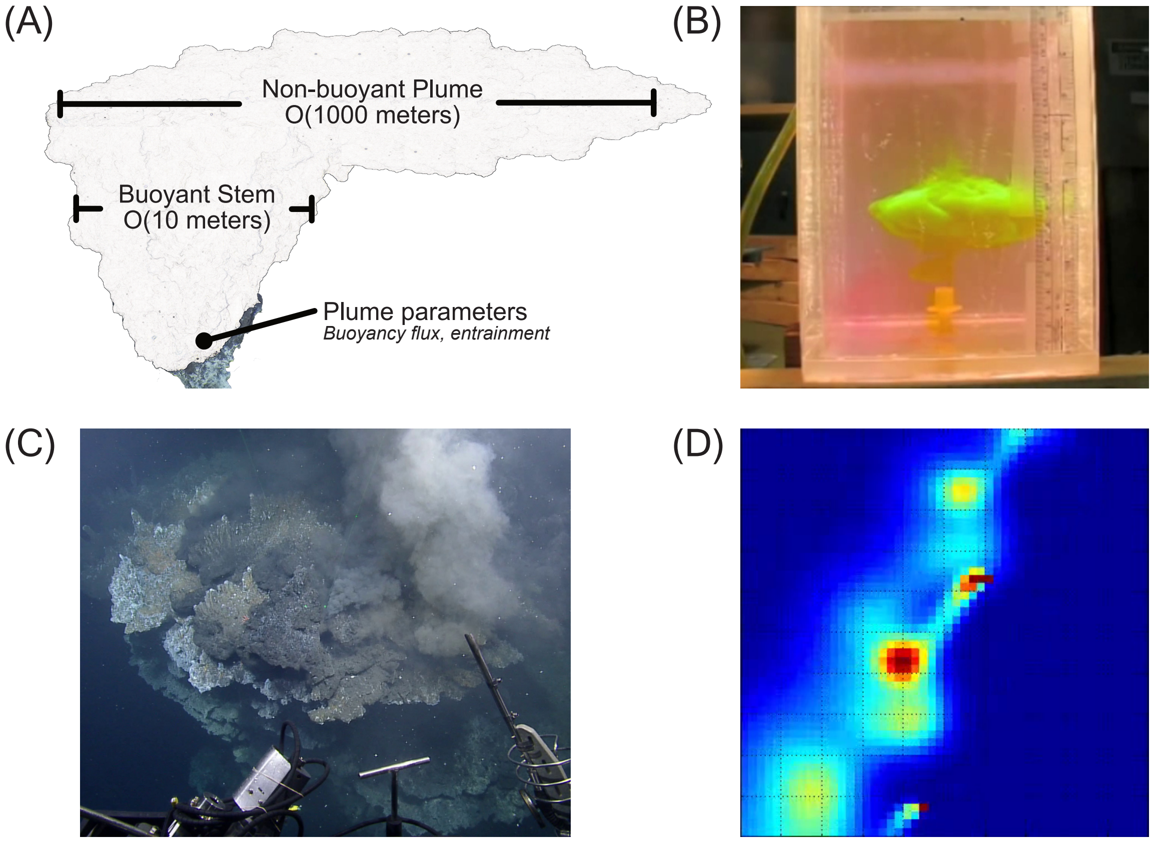


Figure 3.2: ***Deep Sea Hydrothermal Plumes:*** *Four visualizations of the structure of deep sea hydrothermal plumes. (A) Schematic diagram of the structure of a hydrothermal plume (not to scale); (B) The emergence of a buoyant stem and intrusion layer during plume laboratory experiments in a stratified medium (Image credit: Snapshot from [197]); (C) A real-world plume vent source at Guaymas Basin, Gulf of California (Image credit: WHOI JASON team); (D) Simulated multimodal structure of the intrusion layer under advective and diffusive dynamics (Image credit: Michael Jakuba [69])*

from the global maximum. In simulation and field experiments, PLUMES collects more scientifically valuable samples than state-of-the-art planners in a diverse set of environments, with various platforms, sensors, and challenging real-world conditions. This chapter is based on work appearing in Flaspohler et al. [45], Preston [141].

3.1 Related Work

In the following sections, we provide an overview of contemporary related literature.

Informative Path Planning

Within the robotics literature the field of informative path planning (IPP) has combined ideas from experimental design and POMDP planning to develop a suite of heuristic algorithms for sequential decision-making with information rewards on size, weight, and power (SWaP) constrained robotic platforms. IPP algorithms have employed mixed integer linear programs (MILPs) [195], behavior-based reactive controllers [16], traveling salesman formulations [70], branch and bound methods [63], sampling-based motion planners [97], and biomimetic gradient descent algorithms [181] to approximately solve a robotic path planning problem with information rewards. These approaches have been applied to a variety of scientific applications: Hitz et al. [62] demonstrate a surface vehicle for plankton distribution mapping in a lake; Hollinger and Sukhatme [63] use marine vehicles to map the distribution of WiFi signal strength across a lake; Arora et al. [6] use a Bayes net to model the environment and perform hypothesis-driven planning with Monte Carlo tree search for space robotics applications; Manjanna et al. [108, 109] deploy a heterogeneous robotic team to sample chlorophyll density; Karapetyan et al. [81] develop efficient coverage algorithms for river environments; Shkurti et al. [154] use visual sensors to perform underwater following with marine robots; Girdhar et al. [51] develop Bayesian nonparametric models for encouraging novelty seeking in scientific robots. IPP is a large and heterogeneous field. Underlying real-world IPP applications are a host of clever heuristic and algorithmic choices that allow the underlying, information-reward POMDP to be approximated efficiently. Chapter 3 of this thesis continues this line of work and presents an informative path planning algorithm for plume or source seeking applications.

The MSS problem is closely related to IPP problems. Canonical offline IPP techniques for pure information-gathering that optimize submodular coverage objectives can achieve near-optimal performance [19, 165]. However, in the MSS problem, the value of a sample depends on the unknown maximum location, requiring adaptive planning to enable the robot to select actions that explore to localize the maximum and then seamlessly transition to selecting actions that exploitatively collect valuable samples there. Even for adaptive IPP methods, the MSS problem presents considerable challenges. The target environmental phenomenon is partially observable and most directly modeled as a continuous scalar function.

Additionally, efficient maximum sampling with a mobile robot requires consideration of vehicle dynamics, travel cost, and a potentially unknown obstacle map. Handling these challenges in combination excludes adaptive IPP algorithms that use discrete state spaces [6, 100], known metric maps [70, 156], or unconstrained sensor placement [88].

Planning in Continuous Domains

In the MSS problem, the state of the environment can be modeled as a continuous function. PLUMES uses a Gaussian Process (GP) model to represent the belief over this continuous function, and must plan over the uncountable set of possible GP beliefs that arise from future continuous observations. To address planning in continuous spaces, state-of-the-art online POMDP solvers use deterministic discretization [101] or a combination of sampling techniques and particle filter belief representations [89, 155, 159, 173]. Efficiently discretizing or maintaining a sufficiently rich particle set to represent the underlying continuous function in MSS applications is itself a challenging problem, and can lead to inaccurate inference of the maximum [37]. Other approaches have considered using the maximum-likelihood observation to make search tractable [111]. However, this assumption can compromise search and has optimality guarantees only in linear-Gaussian systems [139]. Instead, PLUMES uses Monte Carlo Tree Search (MCTS) with progressive widening, which we call *continuous-observation MCTS*, to limit planning tree growth [34] and retain asymptotic optimality [9].

Rewards and Heuristics

In the MSS POMDP, the reward function is sparse and does not explicitly encode the value of exploration. Planning with sparse rewards requires long-horizon information gathering and is an open problem in robotics [158]. To alleviate this difficulty, less sparse heuristic reward functions can be optimized in place of the true reward, but these heuristics need to be selected carefully to ensure the planner performs well with respect to the true objective. In IPP, heuristics based on the value of information have been applied successfully [62, 88, 111, 171], primarily using the GP-UCB criteria [32, 165]. We demonstrate that within practical mission constraints, using UCB as the heuristic reward function for the MSS POMDP can lead to suboptimal convergence to local maxima due to a mismatch between the UCB heuristic and the true MSS reward. Instead, PLUMES takes advantage of a heuristic function from the Bayesian optimization (BO) community for state-of-the-art black-box optimization [186], which we call *maximum-value information* (MVI). MVI overcomes sparsity and encourages long-term information gathering, while still converging to the true reward of the MSS POMDP.

3.2 The Maximum Seek-and-Sample POMDP

We formalize the MSS problem by considering a target environmental domain as a d-dimensional compact set $\mathbb{X}_w \subset \mathbb{R}^d$. We allow $\mathbb{X}_w$ to contain obstacles with arbitrary geometry and let $\mathbb{X} \subset \mathbb{X}_w$ be the set of reachable points with respect to the robot's initial pose. We assume there is an unknown underlying continuous function $f : \mathbb{X}_w \to \mathbb{R}$, $f \in C^0(\mathbb{X}_w)$ representing the value of a continuous environmental phenomenon of interest. The MSS objective is to find the unique, reachable global maximizer $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathbb{X}} f(\boldsymbol{x})$ by safely navigating while receiving noisy observations of this function f. We assume f is an unknown, black-box function, i.e., we cannot access derivative information or any analytic form.

We model the process of navigating and generating observations as the MSS POMDP: an 8-tuple $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O, b_0, H, \gamma)$, specialized from the general model presented in Sec. 2.5:

- $\mathcal{S}$: a set of tuples consisting of the underlying continuous environment and location of the robot, $\{(f, \boldsymbol{x}) \mid f \in C^0(\mathbb{X}_w), \boldsymbol{x} \in \mathbb{X}\}$.

- $\mathcal{A}$: a discrete set of action primitives available to the agent, e.g., Dubins curves; can be a function of the robot's location $\boldsymbol{x} \in \mathbb{X}$.

- T: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, the transition function, encodes the dynamics of the underlying environment and the robot's dynamics. The environment is assumed to be either static or changing with a random diffusion dynamic (Fig. 3.9); the robot is assumed to transition deterministically given an action.

- R: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the reward of taking action a when problem state is s, i.e., $R(s, a)$.

- $\mathcal{Z}$: a continuous set of possible observations, $\mathcal{Z} \subseteq \mathbb{R}_+$.

- O: $\mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to [0, 1]$, the observation or measurement function, encodes how the robot observes the underlying environment via a noisy point measurement.

- b_0: the initial belief state, $b_0 \in \mathcal{P}(\mathcal{S})$.

- γ: discount factor, $0 \leq \gamma \leq 1$.

The Bellman equation is used to recursively quantify the value of belief $b_t \in \mathcal{P}(\mathcal{S})$ over a finite horizon h under policy π as:

$$V_h^\pi(b_t) = \mathbb{E}[R(s_t, \pi(b_t))] + \gamma \int_{z \in \mathcal{Z}} V_{h-1}^\pi(b_{t+1}^{\pi(b_t),z}) \; \mathbb{P}(z \mid b_t, \pi(b_t)) \; \mathrm{d}z, \tag{3.1}$$

where the expectation is taken over the current belief and $b_{t+1}^{\pi(b_t),z}$ is the updated belief after taking action $\pi(b_t)$ and observing $z \in \mathcal{Z}$. An optimal policy π_h^* over horizon-h is the maximizer of the value function over the space of possible policies Π: $\pi_h^* = \arg\max_{\pi \in \Pi} V_h^\pi(b_t)$. However, Eq. 3.1 is intractable in general continuous state and observation spaces [124]; an optimal policy must be approximated. PLUMES uses a receding-horizon, online POMDP planner and heuristic reward function to approximately solve the MSS POMDP in real-time on robotic systems.

3.3 The PLUMES Algorithm

PLUMES is an closed-loop, online planning algorithm with a sequential decision-making structure:

1. Conditioned on b_t, approximate an optimal policy π_h^* for finite horizon h with $\hat{\pi}_h^*$.
2. Execute the action $a = \hat{\pi}_h^*(b_t)$.
3. Collect observations $z \in \mathcal{Z}$ according to O.
4. Update b_t to incorporate this new observation; repeat.

In the following sections, we define the specific choice of belief model, planning algorithm, and heuristic reward function that PLUMES uses to solve the MSS POMDP.

3.3.1 Gaussian Process Belief Model

We assume the robot's location $\boldsymbol{x}_t$ at planning iteration t is fully observable, and the unknown environmental phenomenon f is partially observable. The full belief-state is represented as a tuple b_t of robot state $\boldsymbol{x}_t$ and environment belief $g_t = \mathcal{P}(f)$ at time t. Because f is a continuous function, we cannot represent the belief g_t as a distribution over discrete states, as is standard in POMDP literature [77], and must choose an alternate representation. PLUMES uses a Gaussian process (GP) [148] to represent g_t conditioned on a history of past observations. Gaussian processes are a powerful tool for representing uncertainty over continuous functions given a set of noisy function observations, and are well suited to represent the environmental state in this application. This GP is parameterized by a mean $\mu(\boldsymbol{x})$ and covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$.

As the robot traverses a location $\boldsymbol{x}$, it gathers observations $z \in \mathcal{Z}$ of f subject to Gaussian sensor noise σ_n^2, such that $z = f(\boldsymbol{x}) + \epsilon$ with $\epsilon \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_n^2)$. Given a history $\mathcal{D}_t = \{\boldsymbol{x}_i, z_i\}_{i=0}^{D-1}$ of D observations and observation locations at planning iteration t, the posterior belief at a new location $\boldsymbol{x}' \in \mathbb{X}$ is computed:

$$g_t(\boldsymbol{x}') \mid \mathcal{D}_t \sim \mathcal{N}(\mu_t(\boldsymbol{x}'), \sigma_t^2(\boldsymbol{x}')), \text{where} \tag{3.2}$$

$$\mu_t(\boldsymbol{x}') = \kappa_t(\boldsymbol{x}')^\top (\mathbf{K}_t + \sigma_n^2 \mathbf{I})^{-1} \boldsymbol{z}_t, \tag{3.3}$$

$$\sigma_t^2(\boldsymbol{x}') = \kappa(\boldsymbol{x}', \boldsymbol{x}') - \kappa_t(\boldsymbol{x}')^\top (\mathbf{K}_t + \sigma_n^2 \mathbf{I})^{-1} \kappa_t(\boldsymbol{x}'), \tag{3.4}$$

where $\boldsymbol{z}_t = [z_0, \ldots, z_{D-1}]^\top$, $\mathbf{K}_t \in \mathbb{S}_+$ is the positive semi-definite kernel matrix with $\mathbf{K}_t[i,j] = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}_t$, and $\kappa_t(\boldsymbol{x}') = [\kappa(\boldsymbol{x}_0, \boldsymbol{x}'), \ldots, \kappa(\boldsymbol{x}_{D-1}, \boldsymbol{x}')]^\top$. This simple, closed-form solution for the posterior predictive distribution enables a planner to predict the value of the environment at a new location and quantify the uncertainty in that prediction. Evaluating (3.3) and (3.4) require inverting the kernel matrix each time a new observation is received, which can become computationally expensive for large observation sets. Several techniques exist for scaling GP models to large, streaming datasets [103]. In Sec. 3.4, our implementation is based on rank-1 update to the kernel matrix that leverages the matrix inversion lemma and is described in App. A.2.

3.3.2 Planning with Continuous-Observation MCTS

PLUMES selects high-reward actions using a receding-horizon search over possible belief states. This search requires a simulator that can sample observations and generate beliefs given a proposed action sequence. For PLUMES, this simulator is the GP model, which represents the belief over the continuous function f, and in turn simulates continuous observations from proposed action sequences by sampling from the Gaussian distribution defined by (3.3) and (3.4).

PLUMES uses continuous-observation MCTS to overcome the challenges of planning in continuous state and observation spaces. Continuous-observation MCTS has three stages: *selection*, *forward simulation*, and *back-propagation*. Each node in the tree can be represented as the tuple of GP belief and robot location, $b_t = (g_t, \boldsymbol{x}_t)$. Additionally, we will refer to

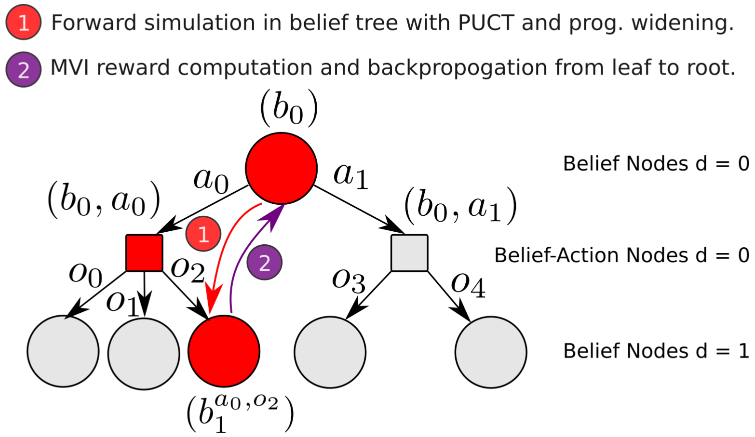

1 Forward simulation in belief tree with PUCT and prog. widening.
2 MVI reward computation and backpropogation from leaf to root.
(b_0)
a_0
a_1
(b_0, a_0)
(b_0, a_1)
1
2
o_0
o_1
o_2
o_3
o_4
$(b_1^{a_0, o_2})$
Belief Nodes d = 0
Belief-Action Nodes d = 0
Belief Nodes d = 1

Figure 3.3: ***Continuous-Observation MCTS:*** *Illustrated to horizon $h = 1$, the tree consists of alternating belief and belief-action nodes. Action decisions are made at belief nodes and random belief transitions according to the observation function occur at belief-action nodes. Note that belief-action nodes have a varying number of children due to progressive widening and unequal simulation (not visualized) due to PUCT policy.*

two types of nodes: belief nodes and belief-action nodes. The root of the tree is always a belief node, which represents the entire history of actions and observations up through the current planning iteration. Through selection and simulation, belief and belief-action nodes are alternately added to the tree (Fig. 3.3).

Selection. From the root, a rollout begins with the *selection* stage, in which a belief-action child is selected according to the Polynomial Upper Confidence Tree (PUCT) policy [9]. The PUCT value $\hat{Q}^*_{aug}(b_t, a)$ is the sum of the average heuristic rewards (i.e., MVI) from all previous simulations and a term that favors less-simulated action sequences:

$$\hat{Q}^*_{aug}(b_t, a) = \hat{Q}^*(b_t, a) + \sqrt{\frac{N(b_t)^{e_d}}{N(b_t, a)}}, \tag{3.5}$$

where $\hat{Q}^*(b_t, a)$ is the average heuristic reward of choosing action a with belief b_t in all previous rollouts, $N(b_t)$ is the number of times the node b_t has been simulated, $N(b_t, a)$ is the number of times that particular action from node b_t has been selected, and e_d is a depth-dependent parameter[1].

Forward Simulation. Once a child belief-action node is selected, the action associated with the child is *forward simulated* using the generative observation model O, and a new belief

[1] Refer to Table 1 of Auger et al. [9] for parameter settings.

node is generated $b_{t+1} = (g_{t+1}, \boldsymbol{x}_{t+1})$ as though the action were taken and samples observed. The simulated observations are drawn from the belief-action node's GP model g_t, and the robot's location is updated deterministically based on the selected action. Since the observations in a GP are continuous, every sampled observation is unique with probability one. *Progressive widening*, with depth-dependent parameter[1] α_d incrementally grows the tree by limiting the number of belief children of each belief-action node. When growing the tree, b_{t+1} is either chosen to be the least visited node if $\lfloor N(b_t, a)^{\alpha_d} \rfloor = \lfloor (N(b_t, a) - 1)^{\alpha_d} \rfloor$, or otherwise is a new child with observations simulated from b_t. By limiting the width of the search tree and incrementally growing the number of explored children, progressive widening avoids search degeneracy in continuous environments.

Back-propagation. Once a sequence of actions has been rolled out to a horizon h, the accumulated heuristic reward is *propagated upward* from the leaves to the tree root. The average accumulated heuristic reward and number of queries are updated for each node visited in the rollout. Rollouts continue until the computation budget is exhausted. The most visited belief-action child of the root node is executed.

Continuous-observation MCTS within PLUMES provides both practical and theoretical benefits. Practically, progressive-widening directly addresses search degeneracy by visiting belief nodes multiple times even in continuous observation spaces, allowing for a more representative estimate of their value. Theoretically, PLUMES can be shown to select asymptotically optimal actions. Analysis in Auger et al. [9] for PUCT-MCTS with progressive widening in MDPs can be extended to PLUMES by reducing the MSS POMDP to an equivalent belief-state MDP [77]. This belief-state MDP has a state space equal to the set of all possible beliefs, and a transition distribution that captures the effect of both the dynamics and the observation model after each action. Planning in this representation is often intractable as the state space is continuous and infinite-dimensional. However, PLUMES plans directly in the belief-state MDP by using its GP belief state to compute the transition function efficiently.

Subsequently, Theorem 1 in Auger et al. [9] shows that for an MDP with a continuous state space, like the belief-state MDP representation suggested, the value function estimated by continuous-observation MCTS asymptotically converges to that of the optimal policy:

$$\left| \hat{Q}_h^*(b_t, a) - Q_h^*(b_t, a) \right| \leq \frac{C}{N(b_t, a)^{\gamma_d}}, \tag{3.6}$$

with high probability [9], for constants $C > 0$ and γ_d*.

3.3.3 Maximum-Value Information Reward

The true reward function for the MSS POMDP would place value on collecting sample points $\boldsymbol{x}$ within an ϵ-ball of the true global maximum $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathbb{X}} f(\boldsymbol{x})$:

$$R((f, \boldsymbol{x}), a) = \mathbb{1}_{\|\boldsymbol{x}-\boldsymbol{x}^*\|<\epsilon}, \tag{3.7}$$

where ϵ is determined by the scientific application. The reward function for the MSS POMDP does not depend on the action $a \in \mathcal{A}$; we will drop this dependence in the remainder of the chapter. Optimizing this sparse reward function directly is challenging. In order to efficiently trade-off between exploration and exploitation, a planner optimizing the MSS reward (3.7) must perform long-horizon search over exploratory actions in order to determine which exploratory actions lead to improvements in expected reward. In practice, evaluating the utility of information gathering actions in a long-horizon POMDP is very challenging and requires significant computational resources [57].

In order to balance exploration and exploitation in the MSS POMDP, PLUMES replaces the true MSS reward with the maximum-value information (MVI) heuristic reward [186]. The belief-dependent MVI heuristic reward $\tilde{R}(b_t, \boldsymbol{x})$ quantifies the expected value of having belief b_t and collecting a sample at location $\boldsymbol{x} \in \mathbb{X}$. MVI reward quantifies the mutual information between the random variable Z, representing the observation at location $\boldsymbol{x}$, and Z^*, the random variable representing the value of the function f at the global maximum:

$$\tilde{R}(b_t, \boldsymbol{x}) = I(\{\boldsymbol{x}, Z\}; Z^* \mid b_t), \tag{3.8}$$

where $Z^* = \max_{\boldsymbol{x}'\in\mathbb{X}} f(\boldsymbol{x}')$. To compute the reward of collecting a random observation Z at location $\boldsymbol{x}$ under belief b_t, we approximate the expectation over the unknown Z^* by sampling from the posterior distribution $z_i^* \sim \mathbb{P}(Z^* \mid b_t)$ and using Monte Carlo integration with M samples [186]:

$$\begin{aligned}\tilde{R}(b_t, \boldsymbol{x}) &= H[\mathbb{P}(Z \mid \boldsymbol{x}, b_t)] - \mathbb{E}_{z'\sim\mathbb{P}(Z^*|b_t)}[H[\mathbb{P}(Z \mid \boldsymbol{x}, b_t, Z^* = z')]], & (3.9)\\ &\approx H[\mathbb{P}(Z \mid \boldsymbol{x}, b_t)] - \frac{1}{M}\sum_{i=0}^{M} H[\mathbb{P}(Z \mid \boldsymbol{x}, b_t, Z^* = z_i^*)]. & (3.10)\end{aligned}$$

Each entropy expression $H[\cdot]$ can be respectively approximated as the entropy of a Gaussian random variable with mean and variance given by the GP equations (3.3) and (3.4), and the entropy of a truncated Gaussian, with upper limit z_i^* and the same mean and variance.

To draw samples z_i^* from the posterior $\mathbb{P}(Z^* \mid b_t)$, we employ spectral sampling [145]. Spectral sampling draws a function $\hat{f}$, which has analytic form and is differentiable, from the posterior belief of a GP with stationary covariance function [61, 186]. To complete the evaluation of Eq. 3.10, $z_i^* \sim \mathbb{P}(Z^* \mid b_t)$ can be computed by applying standard efficient global optimization techniques (e.g., sequential least squares programming, quasi-Newton methods) to find the global maximum of the sampled $\hat{f}$. This results in the following expression for MVI reward [186]:

$$\tilde{R}(b_t, \boldsymbol{x}) \approx \frac{1}{M} \sum_{i=0}^{M} \frac{\gamma_{z_i^*}(\boldsymbol{x}) \phi(\gamma_{z_i^*}(\boldsymbol{x}))}{2\Phi(\gamma_{z_i^*}(\boldsymbol{x}))} - \log(\Phi(\gamma_{z_i^*}(\boldsymbol{x}))) \tag{3.11}$$

where $\gamma_{z_i^*}(\boldsymbol{x}) = \frac{z_i^* - \mu_t(\boldsymbol{x})}{\sigma_t(\boldsymbol{x})}$, $\mu_t(x)$ and $\sigma_t(x)$ are given by (3.3) and (3.4), and ϕ and Φ are the standard normal PDF and CDF respectively. For actions that collect samples at more then one location, the reward of an action $\tilde{R}(b_t, a)$ is the sum of rewards of the locations sampled by that action.

MVI initially favors collecting observations in areas that have high uncertainty due to sampling maxima from the initial uniform GP belief. As observations are collected and uncertainty diminishes in the GP, the sampled maxima converge to the true maximum and reward concentrates locally at this point, encouraging exploitative behavior. This contrasts with the Upper Confidence Bound (UCB) heuristic, which distributes reward proportional to the predictive mean $\mu_t(\boldsymbol{x})$ and weighted variance $\sigma_t(\boldsymbol{x})$ of the current GP belief model (3.3) and (3.4): $\tilde{R}_{\text{UCB}}(b_t, \boldsymbol{x}) = \mu_t(\boldsymbol{x}) + \sqrt{\beta_t}\sigma(\boldsymbol{x})$. As the robot explores, UCB reward converges to the underlying phenomenon, f. The difference in convergence characteristics between MVI and UCB can be observed in Fig. 3.4. Robots optimizing a UCB heuristic will never stop exploring, as reward is distributed throughout the environment; robots optimizing an MVI heuristic will perform only enough exploration to confidently identify the global maximum, and then being to exploit that knowledge to sample near the maximum.

3.4 Experiments and Results

We analyze the empirical performance of PLUMES in an ablation study of MSS scenarios that feature convex and non-convex environments. We compare against three baselines used

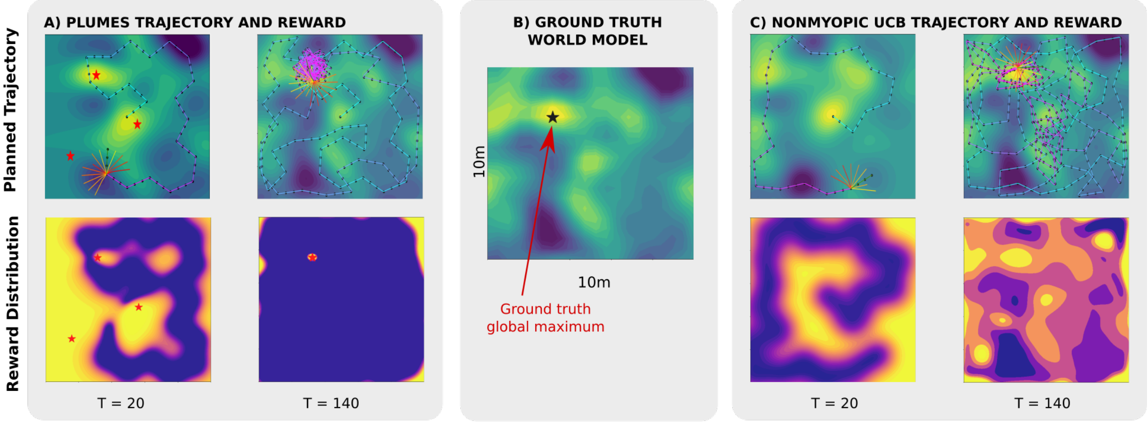


Figure 3.4: ***Convergence of MVI vs UCB Heuristic:*** *The true environmental phenomenon with the global maximum marked by a star is shown in the center; high regions are colored yellow and low regions blue. In (A,C), the robot trajectory and corresponding reward functions are shown early (20 actions) and later (140 actions) in a mission. On the top row, snapshots of the robot belief state with planned trajectories are shown, with recent actions colored pink and earlier actions colored blue. Red stars mark maxima sampled by MVI. In the bottom row, the corresponding reward function is shown, with high-reward regions colored yellow and low reward regions colored purple. By the end of the mission, MVI clearly converges to placing reward only at the global maximum, which in turn leads to efficient convergence of the robot. By contrast, the reward landscape resulting from canonically used UCB converges to the underlying function, causing the UCB planner to uniformly tour high-valued regions of the environment.*

in environmental surveying: non-adaptive lawnmower-coverage (Boustro., an abbreviation of boustrophedonic [29]), greedy myopic planning with UCB reward (UCB-Myopic) [171], and nonmyopic planning with traditional MCTS [25] that uses the maximum-likelihood observation and UCB reward (UCB-MCTS) [111]. The performance of UCB planners has been shown to be sensitive with respect to β value [111]. In order to avoid subjective tuning, we select a time-varying β_t that is known to enable no-regret UCB planning [165, 171]. PLUMES uses continuous-observation MCTS with hyperparameters presented in Auger et al. [9].

To evaluate the mission performance of all planners, we report accumulated MSS reward (Eq. 3.7), which directly corresponds to the number of scientifically valuable samples collected within an ϵ-ball of the true maximum. This metric is reported for all trial scenarios in Tables 3.1 and 3.2. We additionally report several metrics commonly used in IPP to evaluate posterior model quality: overall environmental posterior root mean-squared error (RMSE) and error in posterior prediction of $\boldsymbol{x}^*$ at the end of a mission ($\boldsymbol{x}^*$ error). We use a Mann-

Whitney U non-parametric significance test [110] to report statistical significance (p = 0.05 level) in performance between PLUMES and baseline algorithms.

3.4.1 Bounded Convex Environments

In marine and atmospheric applications, MSS often occurs in a geographically bounded, obstacle-free environment. In 50 simulated trials, we applied PLUMES and our baseline planners to a point robot in a 10 m × 10 m multimodal environment drawn randomly from a GP prior with a squared-exponential covariance function and zero mean function ($l = 1.0$, $\sigma^2 = 100.0$, $\sigma_n^2 = 1.0$ [1%]) (see Fig.3.5). The action set consisted of ten viable trajectories centered at the robot's pose with path length 1.5 m, and samples were collected every 0.5 m of travel. Mission lengths were budgeted to be 200 m. Nonmyopic planners rolled out to a 5-action horizon and were allowed 250 rollouts per planning iteration. Summary simulation results are presented in Table 3.1.

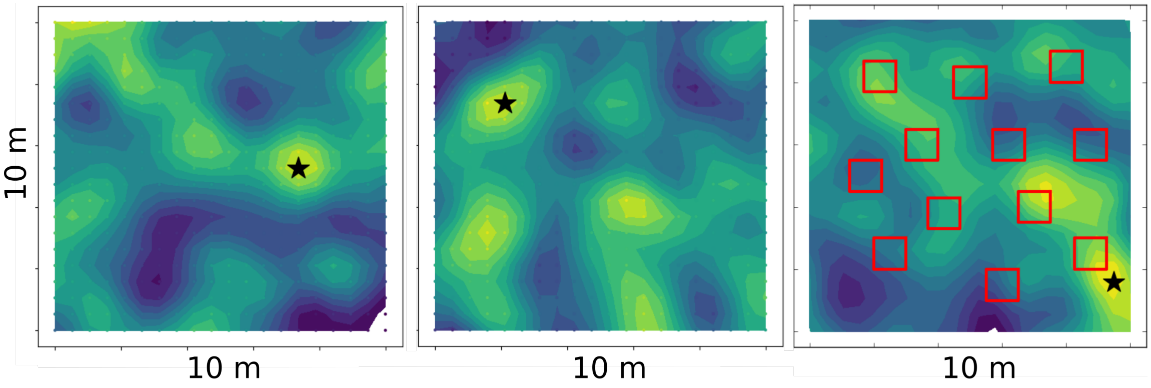


Figure 3.5: ***Simulation Environments:*** *The multimodal simulated 10m × 10m environments. Yellow regions are high-valued; blue regions are low-valued. The global maximum is marked with a star. The left and center environments represent convex-worlds (Section 3.4.1), while the right environment is representative of a non-convex world (Section 3.4.2).*

In these trials, PLUMES accumulated significantly (0.05-level) more reward than baselines. The distribution of accumulated reward (Fig. 3.6) shows that PLUMES has a single dominating mode near reward 200 and few low-performing missions (reward <50). In contrast, both UCB-based methods have distributions which are multimodal, with non-trivial modes in the low-performance region. Boustro. collected consistently few scientifically valuable samples. In addition to collecting many more samples at the maximum, PLUMES achieved statistically indistinguishable levels of posterior RMSE and $\boldsymbol{x}^*$ error compared to

Table 3.1: ***Accumulated True MSS Reward (Eq. 3.7), RMSE, and x^* Error****, Reported as Median (Interquartile Range). Asterisks denote baselines whose difference in performance is statistically significant compared to PLUMES.*

	Convex Simulation Trials $\epsilon = 1.5\,\mathrm{m}$, 50 trials			ASV Trial $\epsilon = 10\,\mathrm{m}$, 1 trial
	MSS Reward	RMSE	x^* Error	**MSS Reward**
PLUMES	**199** (89)	3.8 (9.2)	0.21 (0.23)	**524**
UCB-MCTS	171 (179)*	3.7 (9.6)	0.24 (0.29)	-
UCB-Myopic	148 (199)*	3.6 (9.2)	0.33 (3.25)	-
Boustro.	27 (3)*	2.7 (10.4)	0.26 (0.46)	63

baselines (Table 3.1).

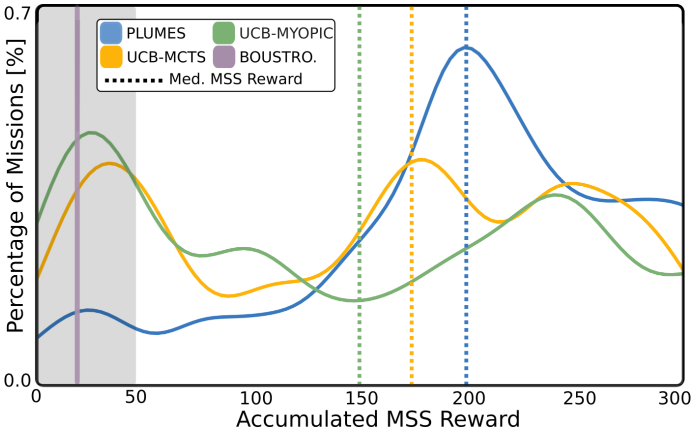


Figure 3.6: ***Distribution of Accumulated MSS Reward in 50 Convex-World Simulations:*** *Accumulated MSS reward is calculated for each trial and the distribution for each planner is plotted as a kernel density estimate (solid line). The dashed lines represent the median accumulated reward for each planner (reported in Table 3.1). The gray area of the plot indicates a low performance region where the planner collected <50 samples near the maximum. PLUMES has a single mode near 200, whereas both UCB-based methods are multimodal, with modes in the low performance region.*

The corresponding field trial for convex-world maximum-search was performed in the Bellairs Fringing Reef, Barbados by a custom-built autonomous surface vehicle (ASV) with the objective of localizing the most exposed coral head Fig. 3.7. Coral head exposure is used to select vantage points for coral imaging [108] and in ultraviolet radiation studies on coral organisms [13]. Due to time and resource constraints, only one trial of two planners was feasible on the physical reef; we elected to demonstrate PLUMES and Boustro., one of the most canonical surveying strategies in marine sciences.

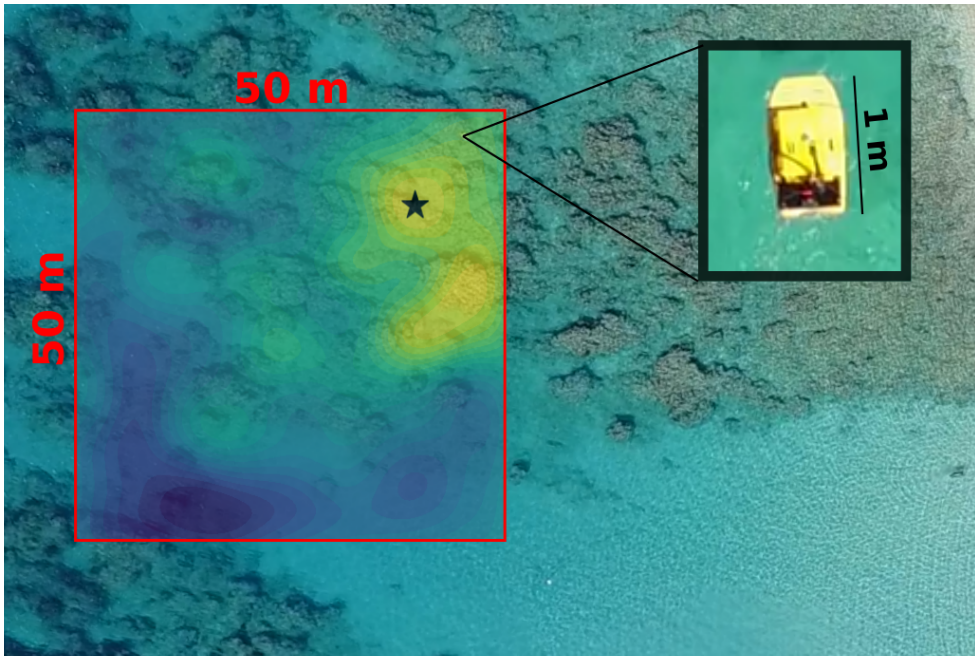


Figure 3.7: ***Coral Mapping with an ASV:*** *The objective of the ASV is to find and sample at the most exposed (shallowest) coral head in a region of Bellairs Fringing Reef, Barbados. Overlaid on the aerial photo is the a priori unknown bathymetry of the region (yellow is shallow, blue is deep). Equipped with an acoustic point altimeter, the ASV must explore to infer the location of the maximum (marked with a star) and then sample at that coral colony.*

The ASV (1 m × 0.5 m) had holonomic dynamics and a downward-facing acoustic point altimeter (Tritech Micron Echosounder) with returns at 1 Hz. Ten dynamically-feasible 10 m straight paths radiating from the location of the ASV were used in the action set. The environment was bounded by a 50 m by 50 m geofence. Localization and control was provided by a PixHawk Autopilot with GPS and internal IMU; the fused state estimate was empirically suitable for the desired maximum localization accuracy ($\epsilon = 10\,\mathrm{m}$). The budget for each mission was 1000 m, which took approx. 45 minutes to travel. The GP kernel was trained on altimeter data from a dense data collection deployment the day before (parameters $l = 2.01$, $\sigma^2 = 0.53$, $\sigma_n^2 = 0.02$ [26%]). Note the high noise in the inferred GP model, as well as the relatively small length-scale in the 2500 m^2 field site. The reconstructed bathymetry and vehicle are shown in Fig. 3.8.

PLUMES successfully identified the same coral head to be maximal as that inferred from the GP trained on prior dense data collection, as indicated by accumulated reward in

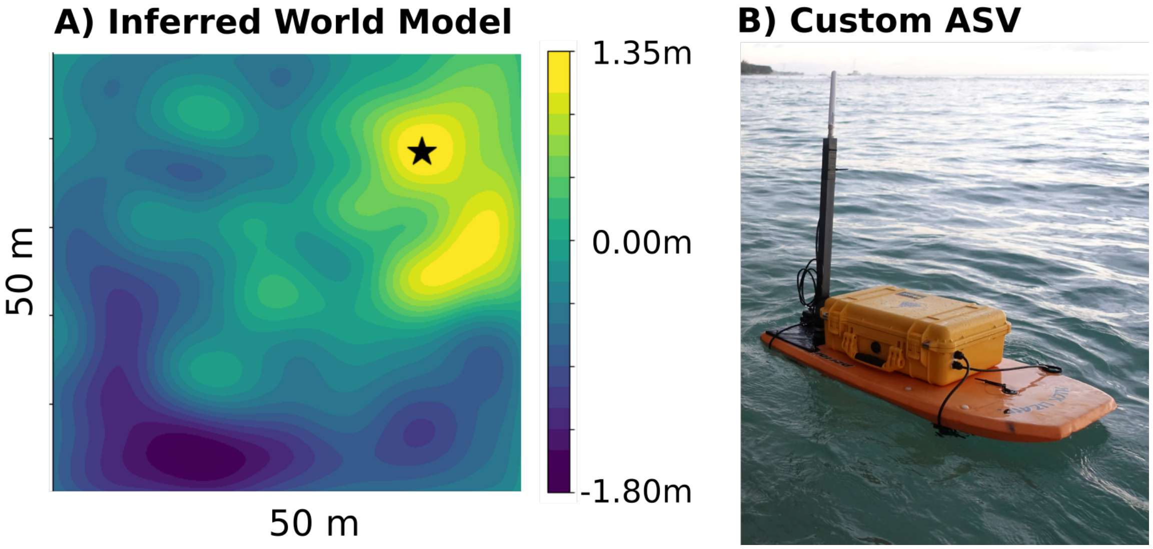


Figure 3.8: ***Coral Head Map and ASV:*** *(A) The ground truth bathymetric map inferred from all collected data, mean corrected in depth. Yellow represents shallower depths, and blue is deeper. The global maximum is marked with a black star. (B) The custom ASV used to traverse the* $2500\,\mathrm{m}^2$ *region.*

Table 3.1, overcoming the challenges of moving in ocean waves, noisy altimeter measurements, and highly multimodal environment. Additionally, the posterior prediction of $\boldsymbol{x}^*$ had an error of only 1.78 m while Boustro. reported 8.75 m error due to its non-adaptive sampling strategy.

In the Bellairs Fringing Reef trials, the environment was assumed to be static. However, in many marine domains the impact of sediment transport, waves, and tides could physically change the location of a maximum over the course of a mission. PLUMES can be extended to dynamic environments by employing a spatiotemporal kernel in the GP model, which allows for the predictive mean and variance to change temporally [157]. If the dynamics of an environment can be encoded in the kernel function, no other changes to PLUMES are necessary; MVI will be distributed according to the time dynamic. Fig. 3.9 demonstrates the properties of PLUMES with a squared-exponential kernel over space ($l = 1.5$, $\sigma^2 = 100$, $\sigma_n^2 = 0.5$) and time ($l = 100$, $\sigma^2 = 100$, $\sigma_n^2 = 0.5$). In this illustrative scenario, the global maximum moved between planning iteration $T = 230$ and $T = 250$. PLUMES with a spatiotemporal kernel maintained multiple hypotheses about the maximum's location given the random-walk dynamic of the environment, resulting in MVI reward being re-distributed between the two maxima over time.

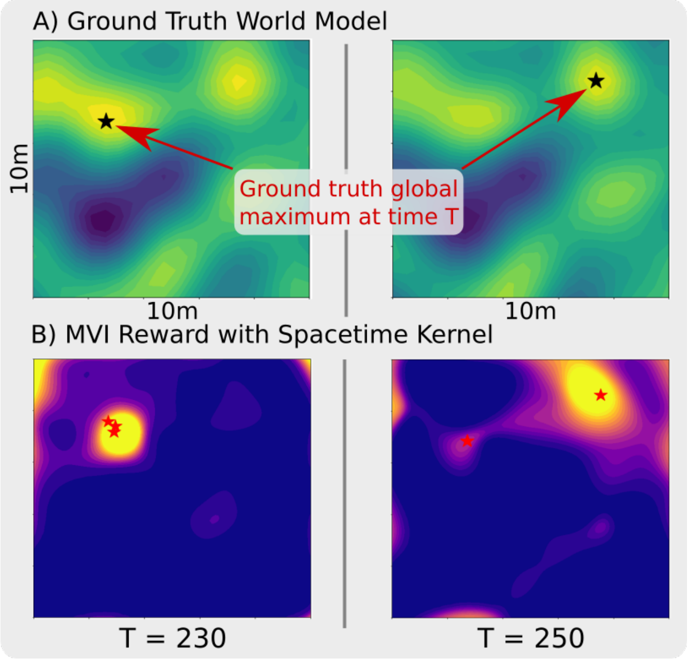


Figure 3.9: ***Extending PLUMES for Spatiotemporal Monitoring:*** *(A) The ground truth map at two planning iterations for a dynamic environment. The maximum is marked with a black star, and migrates from the top left to the top right of the world. (B) MVI reward is redistributed by using a spacetime kernel within PLUMES that captures the environment's dynamics.*

3.4.2 Non-Convex Environments

We next consider non-convex environments with potentially unknown obstacles, a situation that occurs frequently in practical MSS applications with geographical no-go zones for rover or ASV missions, and in indoor or urban settings. We evaluated PLUMES, UCB-Myopic, and UCB-MCTS planners in 50 simulated trials with the same environments, vehicle, and actions as described in Section 3.4.1, with the inclusion of 12 block obstacles placed uniformly around the world in known locations (see Fig.3.5). Boustro. was not used as a baseline because of non-generality of the offline approach to unknown obstacle maps.

As indicated in Table 3.2, PLUMES accumulated significantly more MSS reward than UCB-MCTS and UCB-Myopic, at the 0.05-level. The distribution of reward across the trials is visualized in Fig. 3.11. Like in the convex-world, the PLUMES has a primary mode between reward 200-250, while the UCB-based planners have a primary mode in the low-performance region (reward <50). There was no significant difference between planners

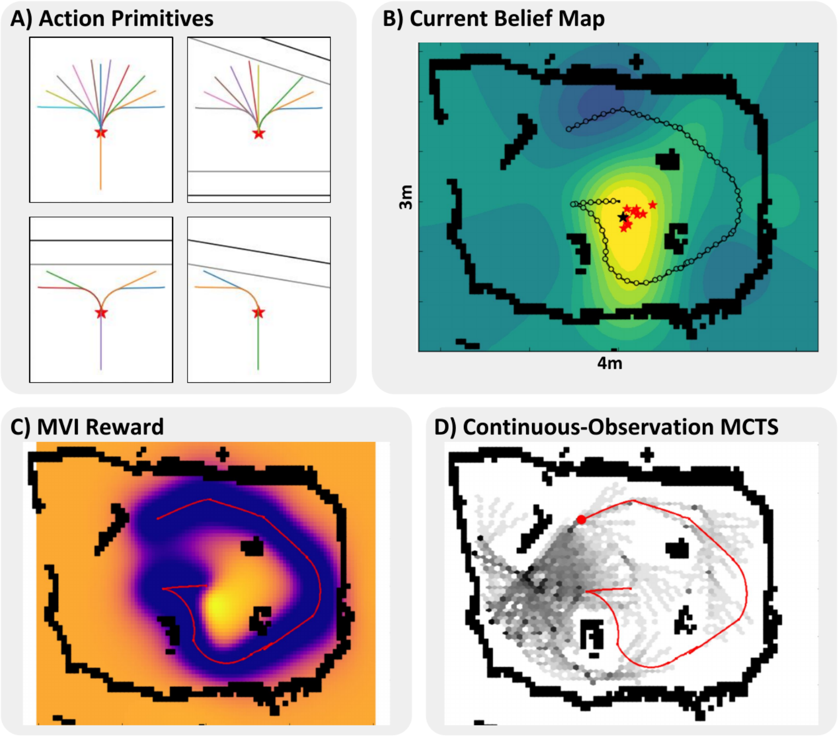


Figure 3.10: ***Snapshot of Unknown Non-Convex Map Scenario:*** *(A) shows examples of how the action-primitives change based upon obstacle detection (black lines) and safety padding (grey lines). (B-D) show a planning iteration of PLUMES, starting with the current belief map and obstacle detections (B). The MVI heuristic is illustrated in (C) where lighter regions are higher value. (D) shows the rollout visibility of continuous-observation MCTS where darker regions are visited more often. Areas of high reward are generally visited more often by the search.*

with respect to RMSE or $\boldsymbol{x}^*$ error. The fact that PLUMES maximized the true MSS reward while achieving statistically indistinguishable error highlights the difference in exploitation efficiency between PLUMES and UCB-based methods.

The simulation experiments assume that a geometric map is known *a priori*. However in practical applications, like indoor gas leak detection, access to a map may be limited or unavailable. We simulate the scenario in which a nonholonomic car equipped with a laser range-finder must build a map online as it seeks the maximum in a cluttered indoor environment (Fig. 3.10). We generate a simulated chemical phenomenon from a GP ($l = 0.8$, $\sigma^2 = 100.0$, $\sigma_n^2 = 2.0$ [2%]), and simulate observations at 1 Hz. The action set for the vehicle consists of eleven 1.5 m Dubins curves projected in front of the vehicle, one straight path behind the vehicle, and a “stay in place” action. Results for five trials are shown in Table

3.2 and illustrate that PLUMES accumulates more MSS reward than baselines, indicating robust performance.

These simulation and robot trials demonstrate the utility of PLUMES compared to canonical and state-of-the-art baselines in a diverse set of environments with challenging practical conditions. For high-stakes scientific deployments, the consistent convergence and sampling performance of PLUMES is critical and beneficial.

Table 3.2: ***Accumulated True MSS Reward (Eq. 3.7), RMSE, and x^* Error****, Reported as Median (Interquartile Range). Asterisks denote baselines whose difference in performance is statistically significant compared to PLUMES.*

	Non-convex Simulation Trials $\epsilon = 1.5\,\text{m}$, 50 trials			Dubins Car Trials $\epsilon = 1.5\,\text{m}$, 5 trials
	MSS Reward	RMSE	$\boldsymbol{x}^*$ Error	**MSS Reward**
PLUMES	**206** (100)	3.6 (2.1)	0.25 (0.56)	**159** (74)
UCB-MCTS	115 (184)*	3.6 (1.5)	0.27 (1.18)	52 (17)
UCB-Myopic	86 (102)*	3.4 (1.0)	0.23 (0.34)	42 (66)
Boustro.	-	-	-	-

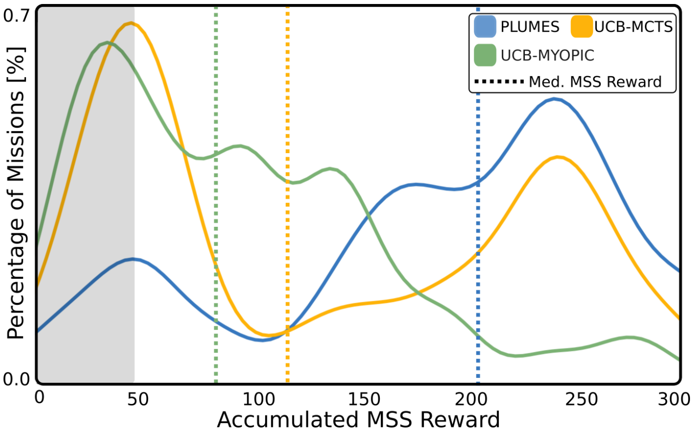


Figure 3.11: ***Distribution of Accumulated MSS Reward in 50 Non-Convex Mission Simulations:*** *Accumulated MSS reward distribution (solid line) and median (dashed line, reported in Table 3.2) for each planner. The gray area of the plot indicates a low performance region (reward <50). PLUMES has few low-performing missions and a primary mode near reward 250. The primary mode of both UCB-based methods is in the low performance region due to convergence to suboptimal local maxima.*

3.5 Conclusion

This work presents maximum-value information as a targeted reward function this is naturally adaptive and avoids a hand-tuned parameter to balance exploration and exploitation. MVI samples potential global maxima from the robot's full belief state to manage exploration and exploitation. In contrast, heuristic functions like UCB place reward on all high-valued or highly uncertain regions, leading to unnecessary exploration and limiting the time available to exploit knowledge of the true maximum. Ultimately, the MVI heuristic allows PLUMES to collect exploitative samples, while still achieving the same overall level of posterior model accuracy (shown by RMSE) as UCB-based planners. Additionally, continuous-observation MCTS allows PLUMES to search over belief-spaces on continuous functions without discretization or maximum-likelihood assumptions. Finally, in dynamic environments for which the dynamics can be encoded into the GP kernel function, PLUMES exhibits continuous monitoring behavior to maintain low uncertainty at the maximum.

Maximum seek-and-sample is a critical task in environmental monitoring and PLUMES presents a solution to this problem with theoretical convergence guarantees, strong empirical performance, and robustness under real-world conditions. One limitation of PLUMES is that it relies on the MVI reward function alone to trade off between exploration and exploitation. In a GP model with squared-exponential or other standard kernel functions, the observations that are most informative about the value of the maximum are also located at the maximum. Because of this property, as the robot explores, MVI concentrates on the maximum location and the robot is drawn there, naturally performing exploitation by sampling at the maximum. This convergence of the exploratory and exploitative objectives need not occur. Using a different underlying model may lead to information about the source being maximized at some location distant from the actual source location. In these models, it would not be sufficient to maximize MVI. The robot would instead need to monitor MVI and intentionally transition to exploitative behavior when the maximum value uncertainty was sufficiently low. The next chapter introduces a more general method for balancing exploration and exploitation in planning problems that directly computes when a transition between exploration and exploitation is necessary.

Chapter 4

Value of Information and Macro-action Discovery in POMDPs

Balancing exploration and exploitation is critical in scientific decision-making. Chapter 3 demonstrates how a careful balance between exploration and exploitation can be struck by using targeted information rewards to perform the minimal amount of exploration necessary to confidently identify a source phenomena in an unknown environment. Generally, performing parsimonious exploration requires understanding how uncertainty in the state of a system or environment affects an agent's ability to perform a given task. Quantifying the sensitivity of task performance to state uncertainty would enable an agent to intelligently trade-off between exploration and exploitation: when task performance is highly sensitive to state uncertainty, an agent can target exploration; when this sensitivity is low, the agent can instead target exploitation.

Quantifying the sensitivity of task performance to state uncertainty for general decision-making problems is challenging, as this sensitivity intimately depends on the specific details of the task and domain of interest. For example, in the hydrothermal plume domain of Chapter 3, the robot's task was to find the global maximum in a multimodal environment. Uncertainty about the state of the environment in regions that were unlikely to contain the global maximum had no impact on task performance. However, if the robot's task was to monitor the boundary between high and low values of the environment — as might be necessary when tracking the progress of, e.g., an oil spill — then knowing the background state of the environment in low-valued regions becomes essential. Additionally, if the robot's sensor is noisy or the underlying environment is dynamic, additional exploration of low-valued regions may be necessary to ensure that these regions do not contain unobserved high-valued points. The solution developed in Chapter 3, in which a specific information metric was designed to target exploration that determined the maximum location in an environment, is not easily adapted to new environments or tasks for general scientific decision-making.

The goal of this chapter is approximate how task performance and model uncertainty are connected for general decision-making problems and then use this to adaptively and efficiently balance exploration and exploitation. We introduce a new metric, termed *value of information (VoI)*, that quantifies task sensitivity to state uncertainty and then use VoI to construct open-loop macro-actions—sequences of exploitative actions—that an agent can execute when improvement in task performance due to exploration is marginal. The motivating scientific application in the chapter is dynamic target tracking for ocean applications, which is introduced in the following section.

Motivating Application: Dynamic target tracking. Traditional marine robotic deployments focus largely on mapping static phenomena in known locations, such as bathymetric features, hydrothermal plume vents, or ice sheets [30, 78, 191]. Due to the immense scale of the marine environment and the relative sparsity of phenomena of interest, it is difficult for scientists to study dynamic phenomena in the ocean, including the long-term behavior of animals, biomass migrations, or other biological or chemical phenomena. Recent innovations in marine robotic platforms has produced small, agile platforms that can track and target dynamic phenomena such as WHOI's Mesobot [196] and the WARP AUV [52] (Fig. 4.1).

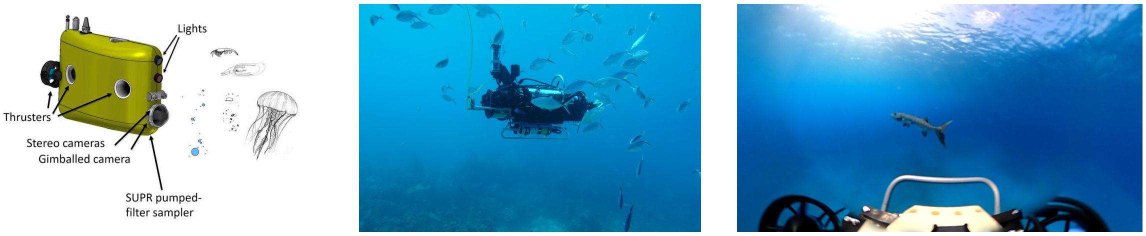


Figure 4.1: ***Dynamic Target Tracking in the Ocean:*** *Examples of ocean platforms that have been recently developed for the task of tracking dynamic biological phenomena in the ocean. Image credit: (Left) WHOI Mesobot Team; (Center/Right) WHOI, WARPLab and WARP AUV.*

These agents use cameras and other sensors to localize and track phenomena of interest as those phenomena locomote and are transported by ocean currents and upwellings. Mesobot, for example, is used to track the diel migration of deep-dwelling animals [196]; the WARP AUV has been used to track fish and other small marine animals in shallow-water environments [52]. These different biological targets have different dynamics, long term behaviors, and can be detected and sensed with different levels of accuracy. These robots must, in turn, understand how the dynamics and observability of each target affects its ability to accomplish the tracking task: if the target moves quickly, unpredictably, and is difficult to detect, the robot must run a high-frequency, closed-loop detector and controller to maintain the target track; if the target moves slowly, predictably, and is easy to detect, the tracking task can be performed even with significant state uncertainty. This motivating application demonstrates how the specifics of a task impact the sensitivity of task performance to state uncertainty.

Overview. The remainder of this chapter introduces a general formulation for quantifying value of information (VoI) in partially observable Markov decision processes (POMDPs). We use VoI to extract belief-dependent, variable-length macro-actions directly from a low-level POMDP model by chaining sequences of exploitative, open-loop actions together when the task-specific value of information (VoI) — the change in expected task performance caused by observations in the current planning iteration — is low. The performance gap between these macro-action policies and the optimal, closed-loop policy is shown to be bounded. This analysis makes use of two classical ideas in POMDP theory: the contractive property of the belief transition operator and the size of a policy's reachable belief space. Finally, a macro-action policy is applied to a simulated version of the dynamic tracking application. In this simulated domain, the robot demonstrates variable exploration policies depending on the predictability and dynamics of the underlying target. This chapter is based on work appearing in Flaspohler et al. [46]

4.1 The Reachable Belief Space and POMDP Hardness

A core challenge in POMDP planning is that searching over possible exploratory actions causes an exponential explosion in the complexity of the planning problem, because the planner must search over possible exploratory actions, simulate received observations, and incorporate those observations into the planner's belief state. Formally, the optimally reachable belief space $\mathcal{R}^*(b_0)$ — the set of beliefs that are reachable from an initial belief b_0 under stochastic observation transitions when following an optimal policy — grows exponentially with the planning horizon in the size of the observation set. The complexity of computing an optimal POMDP policy is related to the covering number of $\mathcal{R}^*(b_0)$ [95], and this exponential growth poses a challenge for planning algorithms that attempt to approximate $\mathcal{R}^*(b_0)$ using offline, point-based approximations [90, 137, 140] or online, sampling methods [155, 159, 172].

To reduce this search complexity, we may want to identify portions of the planning horizon during which an agent can execute a preset, exploitative action and not need to reason over the full dynamics of the belief state. Previous approaches have introduced high-level macro-actions [1] or options [175], such as *drive to the nearest exit*, to reduce planning complexity in complex tasks by executing a fixed, exploitative policy for a given portion of the planning horizon. Policies that use open-loop macro-actions have the dual benefits of a shorter effective planning horizon and smaller reachable belief space (RBS), as a policy's reachable belief space grows linearly rather than exponentially when acting in open-loop (Figure 4.2). However, macro-actions are largely hand-coded [3, 59, 178] or learned without formal guarantees [2, 11, 72]. Here, we address the key challenge of generating macro-actions from a low-level POMDP model such that the resulting policies have bounded regret.

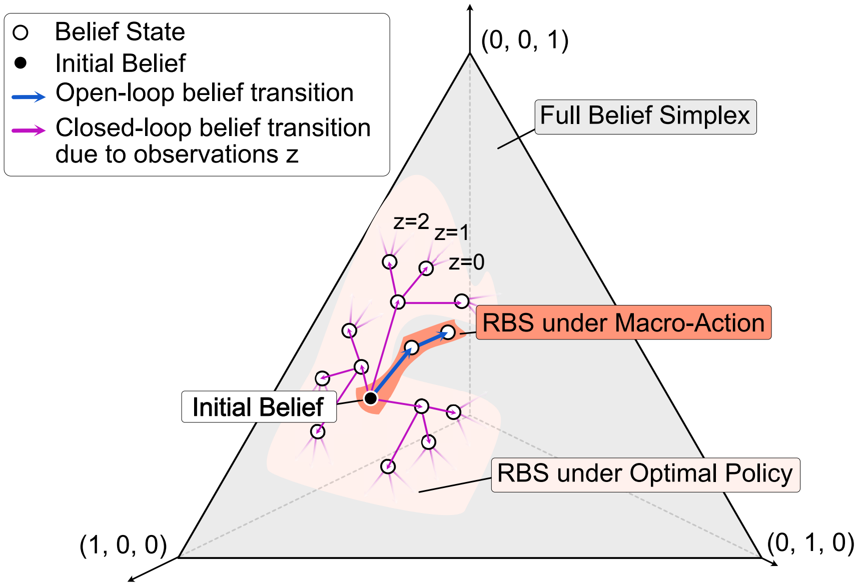


Figure 4.2: ***Reachable Belief Space (RBS) and Macro-actions:*** *POMDP planning algorithms often reason over the value of beliefs in a policy's reachable belief space (RBS). However, the size of a policy's RBS generally grows exponentially with the planning horizon in the size of the observation set $\mathcal{Z}$. This exponential growth is visualized for a three-state discrete POMDP with $|\mathcal{Z}| = 3$. Because the belief transitions deterministically under the open-loop VoI macro-actions, the size of the RBS grows only linearly during macro-action execution.*

This chapter introduces a method for generating belief-dependent, variable-length macro-

[1] We use the term *macro-action* synonymously with open loop action sequence.

actions using a point-based representation of the POMDP value function. Our key insight is to introduce a **value of information (VoI) metric**—which estimates the change in expected task performance caused by sensing in the current planning iteration—and constrain policies to selectively act open-loop when VoI is low. Unlike hand-coded or learned macro-actions, we show that a horizon-H policy utilizing VoI macro-actions has bounded regret r_H compared to the optimal policy. Let V_H^* be the expected reward of an optimal policy and V_H^{MA} the expected reward of the VoI macro-action policy. Then, our main result (Thm. 4.6.2) shows:

$$r_H = \left\|V_H^* - V_H^{MA}\right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}\Big(\delta_{\mathcal{B}}\big(3L + \frac{R_{max}}{1-k\gamma} + L\gamma k\big) + \tau\Big), \tag{4.1}$$

where γ is the POMDP discount factor, L is a Lipschitz constant describing the smoothness of the value function in belief space, and the POMDP reward function is bounded in $[-R_{max}, R_{max}]$.

The three remaining terms — τ, $\delta_{\mathcal{B}}$, and k — elucidate the key trade-offs for task-targeted exploration in POMDP planning. Introducing potentially sub-optimal macro-actions into a policy increases regret. The parameter τ is a VoI threshold, below which the planner acts in open-loop; high values of τ increase macro-action utilization but also increase regret. However, macro-action policies are often easier to approximate than an optimal policy. During planning, we approximate the value function at a set of beliefs that form a $\delta_{\mathcal{B}}$-covering of the macro-action policy's RBS. Since the open-loop belief dynamics are a k-contractive mapping on belief space, this RBS grows slowly when acting in open-loop; macro-action utilization leads to lower values of $\delta_{\mathcal{B}}$ and lower regret bounds. The form of Eq. 4.1 makes the trade-off between exploration and exploitation in POMDP policies explicit. Somewhat surprisingly, although consistent with Eq. 4.1, our empirical results demonstrate that macro-action policies can even outperform approximations of the optimal policy when planning with a finite point-based belief representation.

In the following sections, we introduce VoI macro-action generation and present empirical results in a set of simulated dynamic tracking experiments. Taken together, VoI macro-action generation and the associated regret bound address two fundamental questions for macro-action-based planning in partially observable domains: how do we construct high-value macro-actions and when can we use them without compromising policy performance?

4.2 Related Work

Existing offline [90, 137, 140] and online [155, 159] POMDP planners must contend with the rapid growth of $\mathcal{R}^*(b_0)$ and the resulting difficulty of approximating optimal plans. Previous work has quantified the hardness of approximating optimal POMDP policies in terms of the covering number of $\mathcal{R}^*(b_0)$ [95] and POMDP planners such as SARSOP [90] leverage this insight during planning. Online POMDP solvers, on the other hand, search over a reduced RBS by sampling scenarios in a receding horizon fashion [155, 159, 172]. However, the performance of many online planning algorithms depends on the complexity of the optimal policy [159]. For problems in which the covering number of $\mathcal{R}^*(b_0)$ is large, both offline and online methods have little recourse. By contrast, we explicitly search for near-optimal policies that are easy to approximate by selectively employing open-loop macro-actions to reduce the size of the policy's reachable belief space.

Options and macro-actions [175] have been widely used within the POMDP and reinforcement learning communities to reduce planning complexity. Previous approaches use a prescribed set of macro-actions or closed-loop options, which are identified to provide a useful problem decomposition [3, 59, 65, 84, 99, 178]. These algorithms allow planners to search over shorter effective planning horizons and often benefit from a reduced RBS, but do not provide a mechanism to identify useful options or macro-actions from the underlying planning problem. Recently, work in deep reinforcement learning has attempted to directly learn data-dependent closed-loop options in fully-observable problems [2, 11, 41, 72]. However, these approaches do not provide formal performance guarantees or deal with the growth of the RBS and the other challenges present in partially observable problems.

4.3 Planning Preliminaries

As introduced in previous chapters, a finite-horizon POMDP can be represented as tuple: $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O, b_0, H, \gamma)$, where $\mathcal{S}$ are the states, $\mathcal{A}$ are the actions, and $\mathcal{Z}$ are the observations. At planning iteration t, the agent selects an action $a \in \mathcal{A}$ and the transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ defines the probability of transitioning between states in the world, given the current state s and control action a. After the state transition, the agent receives an observation according to the observation function $O : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{Z})$, which defines the probability of receiving an observation, given the current state s and previous control action a. The reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ serves as a specification of the task. A POMDP is initialized with belief b_0 and plans over horizon H with discount factor γ. In the following, we consider finite-horizon planning problems; extensions of many of the results to discounted infinite-horizon problems is straightforward.

Due to the stochastic and partially observable nature of current and future states, the realized reward in a POMDP is a random variable. Optimal planning is often defined as finding the sequence of policies $\{\pi_t^* : \mathcal{P}(\mathcal{S}) \to \mathcal{A}\}_{t=0}^{H-1}$ that maximize expected reward: $\mathbb{E}\Big[\sum_{t=0}^{H-1} \gamma^t R\big(S_t, \pi_t(b_t)\big) \mid b_0\Big]$, where b_t is the updated belief at time t, conditioned on the history of actions and observations.

The recursively defined horizon-h optimal value function V_h^* quantifies, for any belief b, the expected cumulative reward over the remaining planning iterations when following an optimal policy: $V_0^*(b) = \max_{a\in\mathcal{A}} \mathbb{E}_{s\sim b}[R(s,a)]$ and

$$V_h^*(b) = \max_{a\in\mathcal{A}} \mathbb{E}_{s\sim b}[R(s,a)] + \gamma \int_{z\in\mathcal{Z}} \mathbb{P}(z \mid b, a) V_{h-1}^*(b^{a,z}) \mathrm{d}z \qquad h = 1, \ldots, H-1, \qquad (4.2)$$

where $b^{a,z}$ is the updated belief after taking control action a and receiving observation z, computed via Bayes rule using the transition T and observation O functions. The optimal policy at horizon h is to act greedily according to a one-step look ahead of the value function.

4.4 Generating Belief-Dependent Macro-Actions

In the following section, we introduce value of information (VoI) and describe how VoI can be used to generate belief-dependent, variable-length macro-actions. In (4.4), we define VoI for a given belief as the change in expected long-term reward caused by acting closed-loop and collecting a sensor observation in the current planning iteration. Estimating VoI is critical for selectively employing open-loop macro-actions because *open-loop actions have bounded regret exactly when VoI is low.*

Before presenting the formal definition of VoI, we give an example to provide intuition about when low VoI may arise in planning problems. A belief may have low VoI when:

(i) state dynamics are locally predictable due to the transition function, or

(ii) the reward function is insensitive to uncertainty in the current belief, or

(iii) sensors are locally uninformative or only infrequent sensing is necessary to reduce state uncertainty.

These conditions are visualized schematically in Fig. 4.3. These (and other) conditions arise in many real-world planning problems. Consider the problem of a marine robot tracking a plankton bloom using an ocean flow model. State uncertainty grows slowly when the bloom is localized in regions of near-laminar flow (i). Moderate uncertainty in the bloom location may be tolerable when far from human-occupied beaches (ii). Finally, if the state of the bloom can be observed accurately in certain regions of the ocean, only infrequent observation may be necessary; by contrast, in regions where sensor observations are highly noisy, observations may not meaningfully reduce state uncertainty (iii).

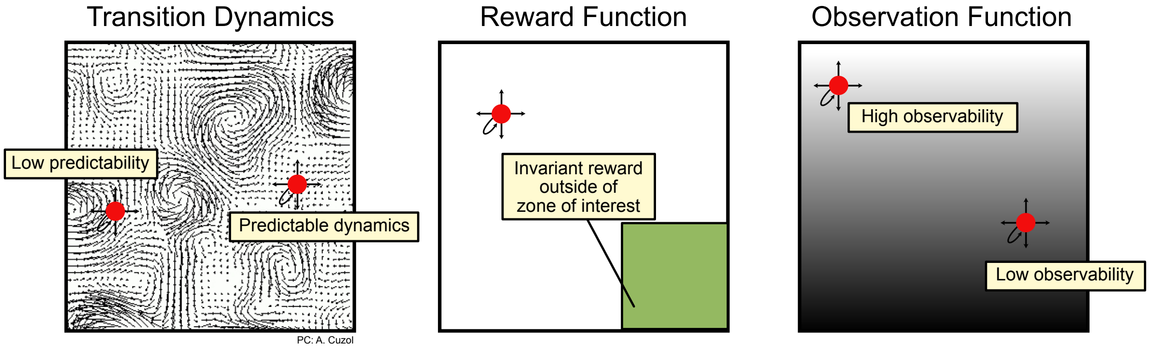


Figure 4.3: ***Conditions for Low VoI:*** *Value of information may be low in a POMDP when the transition dynamics are locally predictable over short horizons (left), the reward function is invariant to aspects of state uncertainty (middle), or when observations are uninformative due to low state observability (right).*

Rather than specialize an algorithm to recognize conditions (i)-(iii), invariably missing other conditions, we find regions of belief-space where open-loop macro-actions are near-optimal by estimating VoI directly using the POMDP value function.

4.4.1 Value of Information for Identifying One-Step Open-loop Actions

We adopt a point-based value function representation, i.e., we approximate the value function using a set of N exemplar beliefs $\mathcal{B} = \{b_i\}_{i=0}^{N-1}$. We compute successive value-function approximations to horizon-H using point-based value iteration [137], where backups of beliefs in the set $\mathcal{B}$ leverage a parametric form of the value function over belief space, such as a set of α-vectors [77] or a deep neural network. We modify the standard value iteration backup operation to compute the VoI, adding open-loop backups whenever the VoI is low. An algorithm summary is presented in Algorithm 1.

We begin by constructing the value function $\hat{V}_h^*$, which approximates the value of a policy that selectively acts in open-loop when VoI is low. $\hat{V}_0^*$ is initialized to the optimal value function. To perform backups of $\hat{V}_h^*$, we compute the open-loop value, V_h^{OLP}, which considers acting in open-loop in the current planning iteration:

$$V_h^{OLP}(b) = \max_{a \in \mathcal{A}} \mathbb{E}_{s \sim b}[R(s,a)] + \gamma \hat{V}_{h-1}^{*}(b^{a,*}) \qquad h = 0, \ldots, H-1, \tag{4.3}$$

where $b^{a,*}$ represents the open-loop belief transition, marginalizing over the received observation.

During the value function backup, we compute both the standard, closed-loop value (Eq. 4.2), denoted V_h^{CLP} and the open-loop value V_h^{OLP}, where for both backups $\hat{V}_{h-1}^{*}$ is used to evaluate the recursion. The difference between the open- and closed-loop value represents the VoI at horizon h for each belief in $\mathcal{B}$ (Figure 4.4).

$$\mathrm{VoI}_h(b) = V_h^{CLP}(b) - V_h^{OLP}(b) \tag{4.4}$$

When VoI is below a regret threshold τ we perform an open-loop backup at b. We then add b to the open-loop set $\mathcal{B}_h^{OLP}$ and store the optimal open-loop action in action set $\mathcal{A}_h^{OLP}$. The resulting backup operator is denoted $\hat{\mathcal{H}}$:

$$\hat{V}_h^{*}(b) = \hat{\mathcal{H}}\hat{V}_{h-1}^{*}(b) = \begin{cases} V_h^{OLP}(b) & \text{if } V_h^{OLP}(b) \geq V_h^{CLP}(b) - \tau \\ V_h^{CLP}(b) & \text{otherwise} \end{cases} \tag{4.5}$$

As shown in Algorithm 1, the belief set $\mathcal{B}$ is initialized at the start of the algorithm by the `construct_belief_set` (e.g., using random beliefs or beliefs reachable under a QMDP policy [102]). As is standard in point-based POMDP literature [137], we perform iterations of alternating value-iteration and belief set updates, in which the value function estimate is used to update the set of reachable beliefs $\mathcal{B}$, via the `expand_belief_set` method, which in turn is used to improve the value function estimate. The algorithm input is a POMDP model, a VoI threshold τ, and a number of belief-set update iterations $iters$.

4.4.2 Chaining Open-loop Actions into Macro-Actions

Value iteration is performed to horizon H using Eq. 4.5, producing the value function $\hat{V}_h^{*}$, sets of open-loop $\mathcal{B}_h^{OLP}$ beliefs, and optimal open-loop actions $\mathcal{A}_h^{OLP}$. We use these sets to generate near-optimal macro-actions for each belief in $\mathcal{B}$ by performing macro-action chaining (Fig. 4.4). An algorithm summary for macro-action chaining is presented in Algorithm 2.

Starting from horizon h, we iterate over beliefs in $\mathcal{B}$. If $b \notin \mathcal{B}_h^{OLP}$, we immediately

```
Input: POMDP = $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O, b_0, H, \gamma)$, $iters$, $\tau$
$\mathcal{B}$ = construct_belief_set($b_0$)
for $i = 0, \ldots, iters - 1$ do
  $\hat{V}_0^*(b) = \max_{a \in \mathcal{A}} \mathbb{E}_{s \sim b}[R(s,a)], \forall b \in \mathcal{B}$
  $\mathcal{B}_0^{OLP} = \mathcal{B}$ // Open loop actions are optimal at horizon-0
  for $h = 1, \ldots, H - 1$ do
    for $b \in \mathcal{B}$ do
      $V_h^{OLP}(b) = \max_{a \in \mathcal{A}} \mathbb{E}_{s \sim b}[R(s,a)] + \gamma \hat{V}_{h-1}^*(b^{a,*})$
      $V_h^{CLP}(b) = \max_{a \in \mathcal{A}} \mathbb{E}_{s \sim b}[R(s,a)] + \gamma \int_{\mathcal{Z}} \mathbb{P}(z \mid b, a) \hat{V}_{h-1}^*(b^{a,z}) \mathrm{d}z$
      $\mathrm{VoI}_h(b) = V_h^{CLP}(b) - V_h^{OLP}(b)$
      if $\mathrm{VoI}_h(b) \leq \tau$ then
        Add $b$ to $\mathcal{B}_h^{OLP}$ and optimal action to $\mathcal{A}_h^{OLP}$.
        $\hat{V}_h^*(b) = V_h^{OLP}(b)$
      else
        $\hat{V}_h^*(b) = V_h^{CLP}(b)$
      end if
    end for
  end for
  $\mathcal{B}$ = expand_belief_set($b_0$, $\mathcal{B}$, $\{\hat{V}_h^*\}_{h=0}^{H-1}$, $\{\mathcal{B}_h^{OLP}\}_{h=0}^{H-1}$)
end for
return $\mathcal{B}$, $\{\hat{V}_h^*\}_{h=0}^{H-1}$, $\{\mathcal{B}_h^{OLP}\}_{h=0}^{H-1}$, $\{\mathcal{A}_h^{OLP}\}_{h=0}^{H-1}$
```

Algorithm 1: Value Iteration for Macro-Action Planning

terminate macro-action chaining. If $b \in \mathcal{B}_h^{OLP}$, we find the associated open-loop action $a \in \mathcal{A}_h^{OLP}$. Critically, the open-loop belief transition is deterministic conditioned on the selected action. Let $b^{a,*}$ be the belief resulting from open-loop action a, which may or may not be included in $\mathcal{B}$. If $b^{a,*} \notin \mathcal{B}$, we evaluate $V_{h-1}^{OLP}(b^{a,*})$ and $V_{h-1}^{CLP}(b^{a,*})$ and use Eq. 4.5 to decide if $b^{a,*} \in \mathcal{B}_{h-1}^{OLP}$. If so, we find the associated optimal action a' and extend the macro-action chain for belief b to include a'; if not, we terminate the chain. We proceed in this manner for the remainder of the planning horizon, or until the deterministically transitioning belief is not in the open-loop set. The chaining process is repeated for each belief in $\mathcal{B}$ and each horizon $h = 0, \ldots, H - 1$.

Macro-action chaining produces a set of belief-dependent macro-actions for beliefs in $\mathcal{B}$. However, during online policy execution, we are likely to encounter beliefs not contained in $\mathcal{B}$. For each belief $b \notin \mathcal{B}$, we execute the macro-action associated with b's nearest neighbor in $\mathcal{B}$ under the L^1 norm. Let V_H^{MA} to denote the expected reward of this approximate

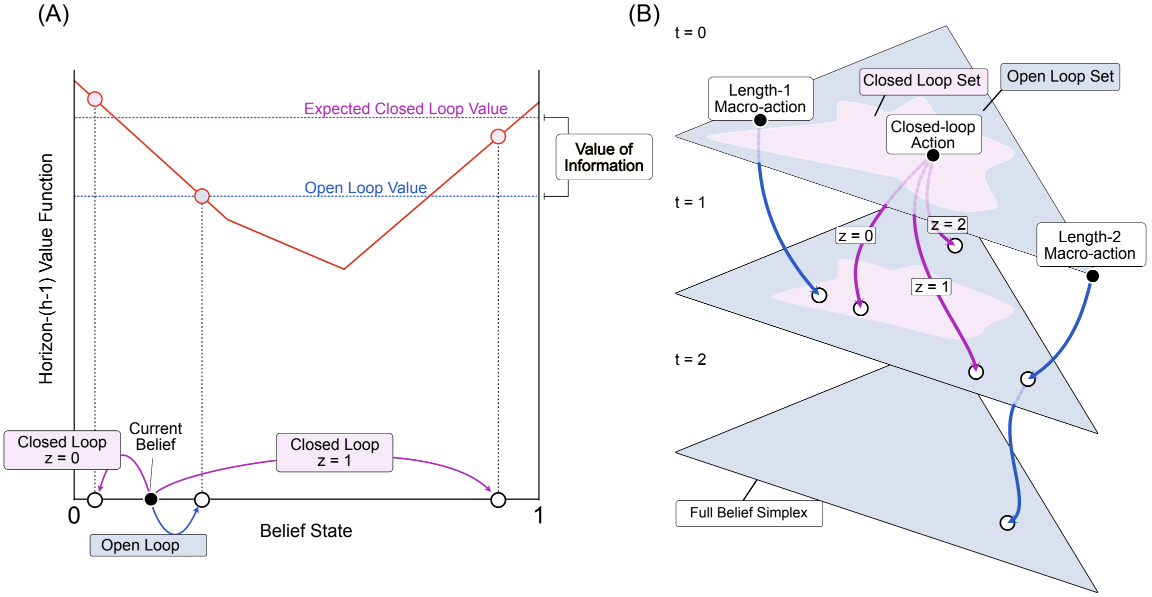


Figure 4.4: ***Macro-action Generation:*** *(A) To compute the value of information at the current belief, we compute immediate reward plus the horizon-$(h-1)$ value under both an open-loop (blue) and closed-loop (purple) belief transition. This difference represents the value of information for long-term task performance. (B) Variable-length macro-actions are constructed by macro-action chaining — for each belief in the open-loop set $\mathcal{B}_h^{OLP}$, we compute the open-loop transition (blue) and terminate macro-action chaining when the belief transitions into the closed-loop set (purple). For beliefs in the closed loop set, no macro-action is generated.*

macro-action policy over horizon H.

Algorithm 2 details the procedure for macro-action chaining. Macro-action chaining takes as input the belief set $\mathcal{B}$, and the VoI macro-action value function $\{\hat{V}_h^*\}_{h=0}^{H-1}$, open-loop sets $\{\mathcal{B}_h^{OLP}\}_{h=0}^{H-1}$, and optimal open-loop actions$\{\mathcal{A}_h^{OLP}\}_{h=0}^{H-1}$. The algorithm checks if a belief b is in the open-loop set at horizon h; for beliefs $b \notin \mathcal{B}$, `check_open_loop` computes VoI as described in Algorithm 1 to determine if an open-loop action is near-optimal at b.

Method Summary VoI macro-action generation first proceeds backward, performing point-based value iteration to estimate VoI and using VoI to decompose the belief space into beliefs for which an open-loop action is near-optimal (open-loop set) and those for which sensing is needed (closed-loop set) (Sec. 4.4.1). Then, macro-action chaining proceeds forward, propagating each belief in the open-loop set forward under the action computed during value iteration and building an open-loop macro-action chain until the propagated belief lies in the closed-loop set (Sec. 4.4.2). The resulting VoI macro-actions are belief-dependent

```
Input: $\mathcal{B}$, $\{\hat{V}_h^*\}_{h=0}^{H-1}$, $\{\mathcal{B}_h^{OLP}\}_{h=0}^{H-1}$, $\{\mathcal{A}_h^{OLP}\}_{h=0}^{H-1}$
macroaction$[b, 0] = \mathcal{A}_0^{OLP}(b), \forall b \in \mathcal{B}$ // Initialize horizon zero macro-actions
for $h = 1, \ldots, H-1$ do
  for $b \in \mathcal{B}$ do
    $b' = b$
    for $t = h, \ldots, 0$ do
      //Check if $b'$ is in the open-loop set and, if so, get open-loop action $a^*$.
      olp, $a^*$ = check_open_loop($b'$, $t$, $\mathcal{B}$, $\{\hat{V}_h^*\}_{h=0}^{H-1}$, $\{\mathcal{B}_h^{OLP}\}_{h=0}^{H-1}$)
      if not olp then
        break // Sensor observation necessary, terminate macro-action chaining
      end if
      macroaction$[b, h]$.append($a^*$)
      $b' = b'^{a^*,*}$ // Transition belief according to open-loop dynamics
    end for
  end for
end for
```

Algorithm 2: Macro-Action Chaining

and variable-length. This method is visualized in Fig. 4.4.

4.5 Macro-actions in Discrete Problems

Sec. 4.4 introduced belief-dependent macro-actions for general POMDP problems by leveraging a representation of the optimal POMDP value function. However, for discrete POMDP problems (problems in which the state, action, and observation spaces are discrete), the value function is known to have special piecewise-linear and convex structure [77], and macro-actions can be generated using this PWLC representation of the value function.

In the following section, we construct an approximation to the optimal PWLC value function by iteratively backing up a set of α-vectors over a finite horizon H. At each backup horizon h, the set of vectors $\hat{\Gamma}_h^*$ contains a mixed set of open-loop and closed-loop vectors.

4.5.1 Value Iteration in Belief Space

Sec. 2.5.1 introduced the PWLC, α-vector formulation for representing the value function of discrete POMDPs. As we saw, the number of α-vectors necessary grows doubly exponentially with planning horizon, making it intractable to form the optimal value function in all but very small problems. One standard method for circumventing this exponential growth is point-based POMDP methods [137]. In point-based approximations, we maintain only the subset of the possible α-vectors that dominate at the exemplar beliefs in $\mathcal{B}$, maintaining the α-vector set at a constant size. Choosing the set $\mathcal{B}$ to be representative of beliefs encountered during policy execution is key in point-based POMDP solvers.

As in Sec. 2.5.1, we will represent the transition function T by a set of transition matrices $\{\mathbf{T}_a\}_{a=0}^{|\mathcal{A}|-1}$, such that $\mathbf{T}_a[i,j] = \mathbb{P}(S_{t+1} = i \mid S_t = j, a_t = a)$. The observation function O will be represented by a set of observation matrices $\{\mathbf{O}_a\}_{a=0}^{|\mathcal{A}|-1}$, such that $\mathbf{O}_a[i,j] = \mathbb{P}(Z_t = i \mid S_t = j, a_{t-1} = a)$.

4.5.2 Open- and Closed-loop α-vectors

As in Sec. 4.4, we construct a value function $\hat{V}_h^*$, which represents the value of beliefs when acting under a policy that selectively leverages open-loop actions. In this discrete setting, $\hat{V}_h^*$ will be represented by a finite set of α-vectors, $\hat{\Gamma}_h^*$.

We initialize $\hat{V}_0^*$ with the set of α-vectors the represent the reward function, $\hat{\Gamma}_0^*$, as described in Eq. 2.11. The backup operator constructs the set $\hat{\Gamma}_h^*$ from the set $\hat{\Gamma}_{h-1}^*$ via the following operations.

Closed-loop First, we construct the standard, closed-loop α-vectors, which represent the value function under closed loop dynamics [77, 137]. First, the one-step reward vectors are constructed for $a \in \mathcal{A}$, which represents the immediate reward of an action a:

$$\Gamma_h^{a,*} = \alpha_a. \tag{4.6}$$

Then, a set of projected α-vectors is constructed, which capture the effect of the observation and transition dynamics on an input belief.

$$\Gamma_h^{a,z} = \{\gamma \alpha_{h-1}^\top \operatorname{diag}(\mathbf{O}_a[z,:])\mathbf{T}_a \mid \alpha_{h-1} \in \hat{\Gamma}_{h-1}^*\} \tag{4.7}$$

Next, for each belief b in the belief set $\mathcal{B}$ the optimal α-vector under an action a for that belief is computed by summing over possible realized observations:

$$\Gamma_h^a = \{\Gamma_h^{a,*} + \sum_{z \in \mathcal{Z}} \operatorname*{arg\,max}_{\alpha \in \Gamma_h^{a,z}} \alpha^\top b \mid b \in \mathcal{B}\}. \tag{4.8}$$

Finally, only the optimal action and its associated α-vector for each belief is maintained:

$$\Gamma_h = \Big\{ \operatorname*{arg\,max}_{\alpha \in \{\Gamma_h^a \mid a \in \mathcal{A}\}} \alpha^\top b \mid b \in \mathcal{B} \Big\}. \tag{4.9}$$

Open-loop To determine the VoI from a specific belief state, we introduce *open-loop α-vectors*, which represent the deterministic transition of belief due to system dynamics in the absence of observations. These open-loop (OLP) α-vectors are constructed similarly to their closed-loop counterparts, where the open-loop transition dynamics are governed by only the transition matrix:

$$\Gamma_h^{a,*,OLP} = \{\gamma \alpha_{h-1}^\top \mathbf{T}_a \mid \alpha_{h-1} \in \hat{\Gamma}_{h-1}^*\} \tag{4.10}$$

$$\Gamma_h^{a,OLP} = \{\Gamma_h^{a,*} + \operatorname*{arg\,max}_{\alpha \in \Gamma_h^{a,*,OLP}} \alpha^\top b \mid b \in \mathcal{B}\} \tag{4.11}$$

$$\Gamma_h^{OLP} = \Big\{ \operatorname*{arg\,max}_{\alpha \in \{\Gamma_h^{a,OLP} \mid a \in \mathcal{A}\}} \alpha^\top b \mid b \in \mathcal{B} \Big\}. \tag{4.12}$$

4.5.3 Value Iteration Backups

At each backup during value iteration, we add a mixture of open- and closed-loop α vectors to our current vector set $\hat{\Gamma}_h^*$. For each belief $b \in \mathcal{B}$, we compute the open- and closed-loop value and the value of information:

$$V_h^{OLP}(b) = \max_{\alpha \in \Gamma_h^{OLP}} \alpha^\top \cdot b \tag{4.13}$$

$$V_h^{CLP}(b) = \max_{\alpha \in \Gamma_h} \alpha^\top \cdot b \tag{4.14}$$

$$\mathrm{VoI}_h(b) = V_h^{CLP}(b) - V_h^{OLP}(b) \tag{4.15}$$

If $\mathrm{VoI}_h(b) \leq \tau$, we add the corresponding open-loop vector to the set $\hat{\Gamma}_h^*$ and add belief b to the open-loop set; otherwise, we add the closed-loop vector. Thus, at horizon h, we represent the value function $\hat{V}_h^*$ using a mixture of open- and closed-loop α-vectors, representing regions of belief space in which open-loop actions are near-optimal

4.5.4 A Note on Algorithmic Complexity

Point-based POMDP algorithms maintain a set of α-vectors of constant size; the main algorithmic cost is construction of the updated set of vectors using the methodology described in the previous section. Traditional point-based methods have complexity $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}||\mathcal{Z}||\Gamma_{h-1}|)$ to generate intermediate α-vectors and complexity $\mathcal{O}(|\mathcal{S}||\mathcal{A}||\mathcal{Z}||\Gamma_{h-1}||\mathcal{B}|)$ to selected the maximizing α-vector for each belief in $\mathcal{B}$ [137]. During value iteration for VoI macro-action generation, evaluating open-loop actions and computing VoI has cost equivalent to adding one additional observation to the observation space that provides no information about the current state. This leads to an algorithmic complexity of $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|(|\mathcal{Z}|+1)|\Gamma_{h-1}|)$ to generate intermediate α-vectors and $\mathcal{O}(|\mathcal{S}||\mathcal{A}|(|\mathcal{Z}|+1)|\Gamma_{h-1}||\mathcal{B}|)$ to evaluate VoI and select the open- and closed-loop sets. For large state spaces, the algorithmic complexity of macro-action generation is dominated by constructing the closed-loop α-vectors during value iteration.

4.6 Macro-Action Policy Analysis

In the following, we show that the macro-action value function V_H^{MA} is within a constant factor of the optimal value function V_H^*.

Let $(\mathcal{S}, \mathcal{F}, \mu)$ be a σ-finite measurable space with σ-algebra $\mathcal{F}$ and measure μ. Let the belief space $\mathcal{P}(\mathcal{S})$ be a subset of $L^1(\mathcal{S}, \mathcal{F}, \mu)$ with L^1 norm $\|\cdot\|_1$ and let $\|\cdot\|$ denote the absolute value on $\mathbb{R}$. Let the point-based belief set $\mathcal{B}$ form a $\delta_{\mathcal{B}}$-covering of a compact set $\mathcal{G} \subseteq \mathcal{P}(\mathcal{S})$ that contains all beliefs reachable under the VoI macro-action policy $\mathcal{R}^{MA}(b_0) \subseteq \mathcal{G}$. We assume the following:

Assumption 4.6.1. *Let V_h^π, the horizon h value function under a policy π be Lipschitz continuous with Lipschitz constant L over the reachable belief space of initial belief b_0 under policy π: $\|V_h^\pi(b_1) - V_h^\pi(b_2)\| \leq L\|b_1 - b_2\|_1, \forall b_1, b_2 \in \mathcal{R}^\pi(b_0)$.*

This assumption holds for many classes of POMDP problems, including finite-horizon discrete POMDPs [77], finite-horizon continuous POMDPS [95], and information reward POMDPs [43]. For example, in the discrete problems presented in Sec. 4.5, the PWLC value function is Lipschitz continuous with respect to the 1-norm on discrete belief states with Lipschitz constant $L \leq \frac{1-\gamma^H}{1-\gamma} R_{\max}$.

Consider the POMDP model $M = (\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O, b_0, H, \gamma)$ with reward bounded in $[-R_{max}, R_{max}]$. We define regret r_H at horizon H as the worst-case difference in long-term expected reward between the optimal policy V_H^* and the approximate VoI macro-action policy V_H^{MA}:

Theorem 4.6.2. *The worst-case regret of a policy using VoI macro-actions with threshold τ is bounded for beliefs in $\mathcal{G}$ by:*

$$r_H = \left\|V_H^* - V_H^{MA}\right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}\Big(\delta_\mathcal{B}\big(3L + \frac{R_{max}}{1-k\gamma} + L\gamma k\big) + \tau\Big). \tag{4.16}$$

We prove Theorem 4.6.2 in the remainder of this section. The proof relies on two key results: Lemma 4.6.1, which bounds value approximation error during VoI-based backups and Lemma 4.6.3, which bounds the error of applying macro-actions computed offline to beliefs at runtime.

4.6.1 Value Backup Error

Let $\mathcal{H}$, V_h^* denote the exact value backup operator and the resulting optimal value function respectively (Eq. 4.2) and $\hat{\mathcal{H}}$, $\hat{V}_h^*(b)$ denote the point-based, macro-action backup and value function (Eq. 4.5), where the optimal open-loop action for each belief is used without macro-action chaining. We show that the error between V_h^* and $\hat{V}_h^*$ is bounded and can be decomposed into error caused by the point-based approximation and error caused by the inclusion of potentially sub-optimal open-loop actions.

Lemma 4.6.1. *The horizon-H value function error caused by including open-loop actions in backups whenever VoI $< \tau$ is bounded for beliefs in $\mathcal{G}$ by $\epsilon_H = \left\|\hat{V}_H^* - V_H^*\right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}(2L\delta_\mathcal{B} + \tau)$.*

This bound illuminates the role of VoI parameter τ in policy performance — for larger τ, open-loop actions are used frequently and $\delta_\mathcal{B}$ will generally decrease, improving policy regret; however, large τ also contributes to the policy regret by allowing open-loop actions be taken even when VoI is high.

Proof. Consider any compact subset β of $\mathcal{G}$; importantly, β can be different from the belief set $\mathcal{B}$ used in planning. We define ϵ_h to be the maximum error in the value function on the set β during the value iteration recursion at horizon h. Let $b_\epsilon \in \beta$ be the belief for which the value function error is maximized and δ be the minimum distance between a belief in $\mathcal{B}$ and b_ϵ: $\delta = \min_{b\in\mathcal{B}}\|b - b_\epsilon\|_1$. Because $\mathcal{B}$ forms a $\delta_\mathcal{B}$ covering of β, we have that $\delta \leq \delta_\mathcal{B}$. We bound ϵ_h (for proof see App. B.1) by the following term: $\epsilon_h = \left\|V_h^*(\beta) - \hat{V}_h^*(\beta)\right\|_\infty \leq 2L\delta_\mathcal{B} + \left\|\mathcal{H}V_{h-1}^*(b) - \hat{\mathcal{H}}\hat{V}_{h-1}^*(b)\right\|$.

The term $2L\delta_{\mathcal{B}}$ represents the value-function error induced by the point-based approximation [137]. We will further examine the term $\mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b)$. Without loss of generality, let a_1 be the optimal, closed-loop action at belief b and a_2 be the near-optimal, open-loop action selected for backing up $\hat{V}^*_h$. Let $\mathcal{H}^{a_1}$ denote the closed-loop value function backup using action a_1 and $\hat{\mathcal{H}}^{a_{2,OLP}}$ denote the open-loop backup using action a_2.

$$\left\|\mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b)\right\| = \left\|\mathcal{H}^{a_1}V^*_{h-1}(b) - \hat{\mathcal{H}}^{a_{2,OLP}}\hat{V}^*_{h-1}(b)\right\|, \tag{4.17}$$

$$\leq \left\|\mathcal{H}^{a_{2,OLP}}V^*_{h-1}(b) + \tau - \hat{\mathcal{H}}^{a_{2,OLP}}\hat{V}^*_{h-1}(b)\right\|, \tag{4.18}$$

$$\leq \left\|\gamma V^*_{h-1}(b^{a_2,*}) + \tau - \gamma\hat{V}^*_{h-1}(b^{a_2,*})\right\| \leq \gamma\epsilon_{h-1} + \tau, \tag{4.19}$$

where if $b^{a_2,*} \notin \mathcal{G}$, we replace $V^*_{h-1}(b^{a_2,*})$, $\hat{V}^*_{h-1}(b^{a_2,*})$ with a valid lower-bound. Expanding the recursion $\epsilon_h \leq \gamma\epsilon_{h-1} + 2L\delta_{\mathcal{B}} + \tau$, we conclude that $\epsilon_H \leq \frac{1-\gamma^H}{1-\gamma}(2L\delta_{\mathcal{B}} + \tau)$. ■

4.6.2 Generalizing Macro-Actions

During policy execution, we generalize macro-actions computed for beliefs in $\mathcal{B}$ to new beliefs. The error induced by this approximation can be bounded by demonstrating that the open-loop dynamics are a non-expansive mapping in belief-space, ensuring that during an open-loop macro-action, the distance between the forward-propagated beliefs can be no larger than their initial separation δ.

Lemma 4.6.2. *(Lasota and Mackey [93]) The open-loop dynamics are a non-expansive mapping in belief space. Consider two beliefs $b_1, b_2 \in \mathcal{P}(\mathcal{S})$ such that $\|b_1 - b_2\|_1 = \delta$. Then, for any action a taken in open-loop, it follows that $\left\|b_1^{a,*} - b_2^{a,*}\right\|_1 \leq k\delta$ for $0 \leq k \leq 1$.*

Let V^{MA}_h denote the value of following an approximate macro-action policy from belief b at horizon h, where $b \notin \mathcal{B}$ and the macro-action computed for its nearest neighbor b_* is instead executed.

Lemma 4.6.3. *The additional value function error of approximating the VoI macro-action at belief b using its nearest neighbor b_* under k-contractive open-loop dynamics is bounded by:*

$$\eta_H = \left\|\hat{V}^*_H - V^{MA}_H\right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}\Big(L\delta_{\mathcal{B}} + \frac{R_{max}\delta_{\mathcal{B}}}{1-\gamma k} + L\gamma k\delta_{\mathcal{B}}\Big). \tag{4.20}$$

Proof. Consider any compact subset β of $\mathcal{G}$. Then $\eta_h = \left\|\hat{V}^*_h(\beta) - V^{MA}_h(\beta)\right\|_\infty$.

Without loss of generality, let b be the belief for which η_h is maximized, let b_* be its nearest neighbor in $\mathcal{B}$, and let $A_l = \{a_1, \dots, a_l\}$ be the length-l macro-action that is optimal at b_*. We show (details in App. B.1) that η_h can be decomposed into error incurred during the macro-action and error over the remainder of the planning horizon. We use the notation $b^{A_{1:i}}$ to denote an updated belief after taking the first i actions of macro-action A_l in open-loop.

$$\eta_h \leq L\delta_{\mathcal{B}} + \left\| \sum_{i=0}^{l-1} \gamma^i \Big(\mathbb{E}_{s \sim b_*^{A_{1:i}}}[R(s, a_i)] - \mathbb{E}_{s \sim b^{A_{1:i}}}[R(s, a_i)] \Big) \right\| + \gamma^l (Lk^l \delta_{\mathcal{B}} + \eta_{h-l}). \tag{4.21}$$

The form of Eq. 4.21 reflects the expected reward when following the macro-action A_l from both belief b and b_* and then reverting to the macro-action policy from the resulting belief. We bound Eq. 4.21 by application of the non-expansive property:

$$\left\| \sum_{i=0}^{l-1} \gamma^i \Big(\mathbb{E}_{s \sim b_*^{A_{1:i}}}[R(s, a_i)] - \mathbb{E}_{s \sim b^{A_{1:i}}}[R(s, a_i)] \Big) \right\|, \tag{4.22}$$

$$\leq \sum_{i=0}^{l-1} \gamma^i R_{max} \left\| b_*^{A_{1:i}} - b^{A_{1:i}} \right\|_1 \leq \sum_{i=0}^{l-1} \gamma^i R_{max} k^i \| b_* - b \|_1 \leq \frac{1 - \gamma^l k^l}{1 - \gamma k} R_{max} \delta. \tag{4.23}$$

Plugging this expression into Eq. 4.21, we have the recursion: $\eta_h \leq L\delta_{\mathcal{B}} + \frac{1-\gamma^l k^l}{1-\gamma k} R_{max} \delta_{\mathcal{B}} + \gamma^l L k^l \delta_{\mathcal{B}} + \gamma^l \eta_{h-l}$. This expression depends on l, the length of the optimal macro-action at horizon h, in a complex way. Because l is variable and unknown *a priori*, we replace l with its worst-case value in each expression: $\eta_h \leq L\delta_{\mathcal{B}} + \frac{R_{max}\delta_{\mathcal{B}}}{1-\gamma k} + \gamma L k \delta_{\mathcal{B}} + \gamma \eta_{h-1}$ and expand the recursion. ■

Analysis Summary To bound the regret of the VoI macro-action policy compared to the optimal policy, we first bound the error caused by using sub-optimal open-loop actions when VoI was below a threshold τ. We then bound the regret of generalizing macro-actions generated for beliefs in $\mathcal{B}$ to new beliefs encountered during policy execution. Finally, we combine these approximation errors:

Proof. (Theorem 4.6.2) We bound the regret of the VoI macro-action policy on the set $\mathcal{G}$ as follows:

$$r_H = \left\|V_H^* - V_H^{MA}\right\|_\infty \leq \left\|V_H^* - \hat{V}_H^*\right\|_\infty + \left\|\hat{V}_H^* - V_H^{MA}\right\|_\infty = \epsilon_H + \eta_H. \tag{4.24}$$

The result follows by applying Lemma 4.6.1 and Lemma 4.6.3 to bound ϵ_H and η_H. ■

4.7 Experiments

We present experimental results designed to highlight various aspects of VoI macro-actions and provide insight into the nature of the regret bound and its implications macro-action design. We assume discrete states, actions and observations and represent the value function by a piecewise-linear and convex (PWLC) collection of α-vectors [77]. An adaptation of the algorithm presented in Section 4.4 to a PWLC value function is contained in Sec. 4.5. We use this example to expose several key low VoI regimes caused by structure in the system dynamics, reward function, and transition dynamics. We present the following two experiments.

Experiment 1: Predictable Dynamics Our first experiment is designed to highlight how locally predictable state dynamics can lead to low VoI values (condition (i), Sec. 4.4). We test both a *random-walk* target dynamic and a *boundary* target dynamic in which the target performs a random walk in the interior but moves deterministically clockwise on the boundaries, with probability $\alpha = 0.20$ of returning to the interior (Fig. 4.5). These dynamics exemplify different VoI regimes. In the boundary dynamic, VoI is low on the boundary, allowing for long macro-actions. In the random walk dynamic, VoI is more uniform over belief space; only short macro-actions are possible before sensing is necessary.

Experiment 2: Invariant Reward Our second experiment introduces conditions (ii) and (iii) (Sec. 4.4) by rewarding the agent for tracking the target only in a single zone of interest in the upper left corner of the world and imposing non-uniform observation noise inspired by the Dark-Light POMDP problem [139, 172], such that the agent can sense the target's location most accurately on the bottom half of the world (Fig. 4.8). The agent follows a boundary transition dynamic (condition (i)). The agent must perform information gathering – moving to the bottom of the world to localize the target — before returning to the upper corner to track the target in the zone of interest. VoI is high when the target is nearing the zone of interest and can be localized with sufficient accuracy via sensing.

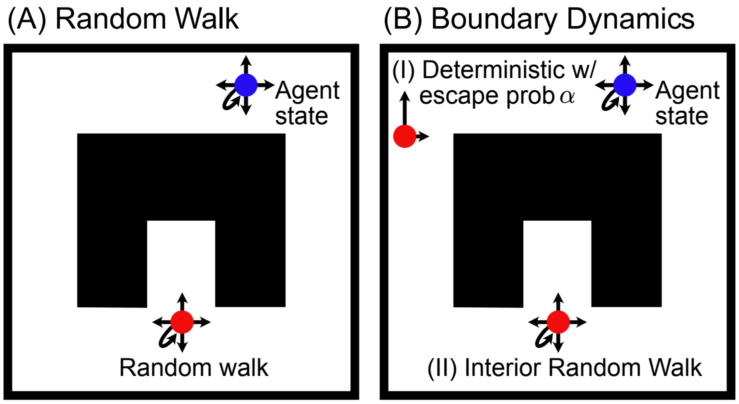


Figure 4.5: ***Dynamic Tracking Experiment:*** *An agent (blue) tracks a partially observable target (red) in an environment with obstacles and boundaries (black) under (A) a random walk or (B) a boundary dynamic with escape probability α.*

4.7.1 Experimental Setup and Parameters

We demonstrate macro-action generation in a dynamic tracking problem (Fig. 4.5), in which a fully observable, actuated agent tracks a partially observable target moving in a known 10×10 discretized map ($|\mathcal{S}| = 10,000$). The agent can move in each of the cardinal directions or stay in place ($|\mathcal{A}| = 5$). The agent observes the target location with discretized Gaussian noise (diagonal covariance, $\sigma^2 = 6.25$, $|\mathcal{Z}| = 100$) in Experiment 1 and discretized Gaussian noise with diagonal covariance varying linearly depending on the agent's location, from $\sigma = 0$ (perfect observation) in the bottom row of the world to $\sigma^2 = 25.0$ in the top row of the world in Experiment 2. We plan over a horizon of 75 iterations for Experiment 1 and 25 iterations for Experiment 2. We have discount factor $\gamma = 0.99$. In Experiment 1, the reward function penalizes the squared Euclidean distance between the target and the agent, and in Experiment 2, the agent receives a reward of 50.0 if it is in the cell as the target when the target is in the zone of interest ($\{0, 1\} \times \{0, 1\}$); otherwise, the agent receives zero reward. The VoI threshold τ is set to $\tau = 5$ in Experiment 1 and $\tau = 0.05$ in Experiment 2.

Policies are constructed using point-based value iteration [137] with a fixed-size belief set $|\mathcal{B}| = 285$ and executed in a set of $M = 500$ tracking experiments for evaluation. The belief set $\mathcal{B}$ is initialized to beliefs reachable under a QMDP policy [102] and three iterations of alternating value-iteration and belief set updates are performed, in which the value function estimate is used to update the set of reachable beliefs $\mathcal{B}$, which in turn is used to improve the value function estimate [137]. These values were determined by computational constraints;

value function convergence during experimentation was not assessed. Additionally, although PBVI formed the base approximation algorithm for these experiments, any approximation of the POMDP value function can form the base of macro-action construction. Further improvement may be seen if algorithms such as SARSOP [90] that explicitly attempt to approximate the optimally reachable belief space are used as the base approximation.

The VoI macro-action policy (VoI MA) is compared against an approximation to the closed-loop optimal policy (*base closed-loop*, Base CL) and a *fixed length macro-action* (Fixed MA) policy, which is constrained to act closed-loop only every $T = 15$ planning iterations. For all three policies, value function approximation is performed using a custom implementation of PBVI [137].

4.7.2 Experimental Results

Results for Experiment 1 are shown in Table 4.1. Under the boundary dynamic, the VoI macro-action policy has higher cumulative reward than the Base CL and Fixed MA policies. This may seem counterintuitive — the performance of the optimal policy is an upper bound on the VoI macro-action policy. However, this is an example of the trade-off between policy complexity and approximability indicated by Theorem 4.6.2. The observed value of $\delta_{\mathcal{B}}$ is significantly lower for the VoI macro-action policy (Table 4.1), indicative of the macro-action policy's smaller reachable belief space. We note that the fixed macro-action policy has the smallest value of $\delta_{\mathcal{B}}$. However, the fixed-length macro-actions, like many other hand-coded macro-actions, can be arbitrarily sub-optimal. For the random walk dynamic, there is less opportunity to exploit open-loop actions and we see that, as we would hope, VoI MA policy performance reverts to that of the closed-loop policy.

	Boundary Dynamics, $\alpha = 0.20$		Random Walk Dynamics	
Planner	Total Reward	Empirical $\delta_{\mathcal{B}}$	Total Reward	Empirical $\delta_{\mathcal{B}}$
Base CL	2800.2 (465.6)*	0.45 (0.04)*	3035.8 (257.5)	0.42 (0.04)*
VoI MA	**3039.5 (274.4)**	**0.29 (0.03)**	**3050.8 (212.9)**	**0.31 (0.04)**
Fixed MA	2584.3 (460.5)*	0.17 (0.02)*	2946.6 (311.4)*	0.15 (0.02)*

Table 4.1: ***Results for Experiment 1:*** *Realized reward (higher is better) and empirical estimates of $\delta_{\mathcal{B}}$ (lower is better) during $M = 500$ experiments using VoI MA, Base CL, and Fixed MA policies (mean, std). Significant differences from VoI MA (two-sided Welch's t-test with Bonferroni correction, $p < 0.05/2$) are indicated with an asterisk.*

We additionally explore the effect of the VoI threshold τ on planner performance, macro-action utilization, and the value of $\delta_{\mathcal{B}}$. Results are presented in Fig. 4.6. As τ increases, the VoI macro-action policy acts in open-loop for a larger fraction of the planning horizon and the value of $\delta_{\mathcal{B}}$ decreases. The realized reward reflects the balance between these two terms, initially increasing as $\delta_{\mathcal{B}}$ decreases, before finally decreasing as the policy incorporates more sub-optimal open-loop actions.

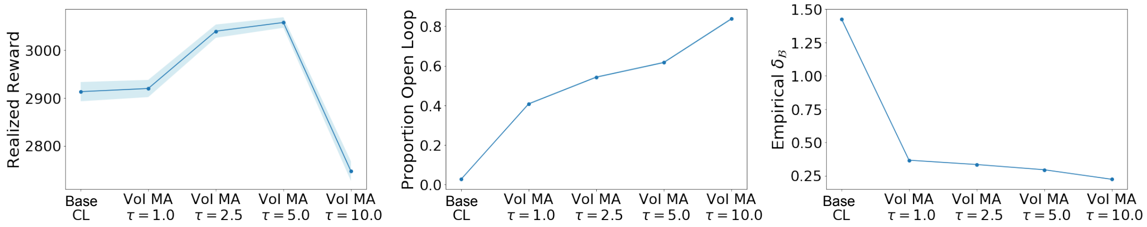


Figure 4.6: ***Macro-action Discovery Results:*** *(Left) Realized reward under Base CL and VoI macro-action policies with increasing values of the VoI threshold τ. (Center) The proportion of the planning horizon for which open-loop macro-actions are employed. (Right) The empirical value of $\delta_{\mathcal{B}}$. Plots show mean and standard error in $M = 500$ trials.*

We can further explore the planner behavior in the random walk and boundary dynamic experiments presented in Table 4.1. Fig. 4.7 provides scatter plots comparing the performance of the VoI macro-action policy and the best closed-loop policy. Each point in the scatter plot represents a paired experiment with identical target dynamics. These plots highlight the variability of the best closed-loop planner and the reduction in the empirical value of $\delta_{\mathcal{B}}$ under the VoI macro-action policy. Fig. 4.7 additionally presents histograms showing the proportion of the planning horizon spent in open loop and the length of macro-actions taken by the agent for each dynamic under the VoI macro-action policy. These plots highlight the utility of using VoI to selectively act in open loop; the VoI macro-action policy acts in open-loop for a large fraction of the planning horizon, using extended sequences of open-loop actions, and yet performs at least as well as the best closed-loop planner.

For Experiment 2, we visualize the length of the discovered belief-dependent macro-actions. To visualize macro-actions as a function of the high-dimensional belief space, for each possible state in the world, we compute the length of the macro-action corresponding to a belief where the target is localized to that state (the target probability mass function is a Dirac delta function), averaged over all possible corresponding states of the agent. Results are shown in Fig. 4.8. The effect of structure in the POMDP model on the discovered macro-actions is evident — when the target is localized in the zone of interest (upper left corner) or has just passed the zone and is moving clockwise due to the boundary dynamic,

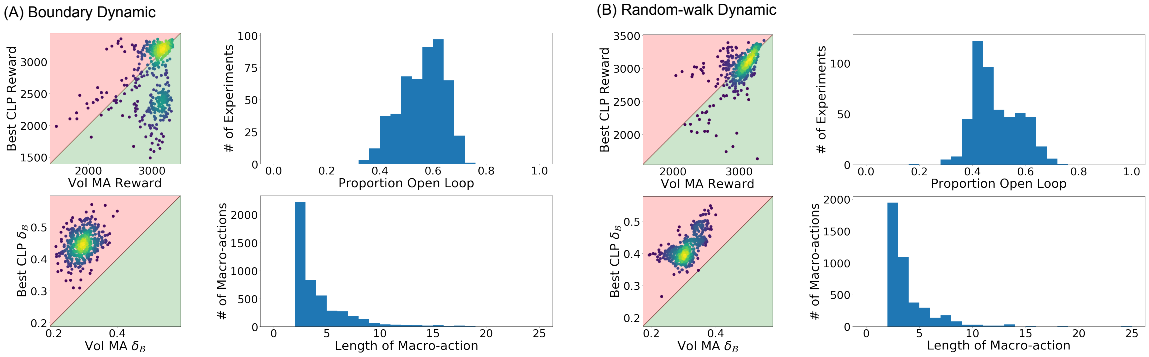


Figure 4.7: ***Macro-actions in (A) Boundary versus (B) Random Walk Dynamics****: (Left) Performance for the VoI MA policy versus the Best CL policy in paired experiments. In the top row, points in the green region represent experiments for which VoI MA outperforms Best CL, achieving higher realized reward. In the bottom row, points in the red region represent experiments for which VoI MA outperforms Best CL, achieving lower values of* $\delta_{\mathcal{B}}$. *Color indicates point density (yellow high density to blue low density). (Right) Histograms of the proportion of the planning horizon the VoI MA planner spends in open loop and the length of all executed VoI macro-actions across* $M = 500$ *experiments.*

the agent can achieve high reward with open loop macroactions; when the target is in the lower half of the world and moving towards the zone of interest, sensing is crucial. In this experiment, the VoI MA policy achieves higher average reward (18.69) than the Base CL (15.08) and Fixed MA (13.03) policies over $M = 100$ simulated trails.

4.8 Conclusion

This work presents value-of-information (VoI) as a novel metric for quantifying the benefit of exploration or observation in a sequential decision-making task. When VoI is low, further sensing (exploration) will not improve task performance and a planning agent should favor action based on its current measurements and model (exploitation).

We leverage VoI to design open-loop macro-actions for planning under uncertainty. By generating macro-actions using VoI, we can bound the regret of macro-action policies with respect to the optimal, closed-loop policy. Leveraging open-loop macro-actions within POMDP policies can reduce the size of a policy's reachable belief space and thus the complexity of planning. This has direct implications for the performance of point-based POMDP policies, as we show theoretically in Theorem 4.6.2 and experimentally in a set of dynamic tracking experiments. VoI macro-actions balance the planning complexity induced

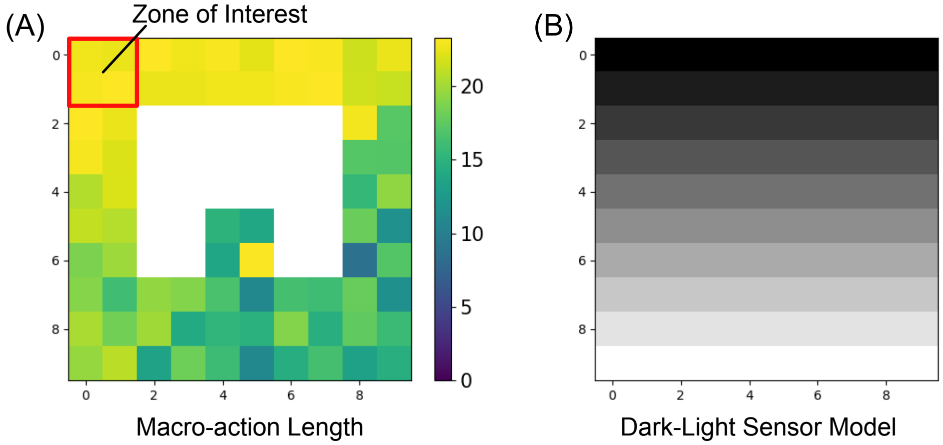


Figure 4.8: ***Visualizing Belief-Dependent Macro-Actions:*** *(A) For each state in the world (shown spatially), the average length of the discovered VoI macro-action is denoted by color for each possible state of the target. The bugtrap obstacle is shown in white. If the target is localized near the upper-left corner of the world, the agent can confidently track the target in the zone of interest, and long-horizon macro-actions are possible. Sensing is more important in the bottom half of the world; the agent must move to the higher observability regions near the bottom of the world in order to reduce target uncertainty before it enters the zone of interest. VoI macro-actions are also sensitive to the state dynamics — for example, long-horizon macro-actions are possible when the target is trapped in the bugtrap obstacle. (B) The dark-light sensor model; sensor noise increases linearly from the bottom (light) to the top (dark) of the world.*

by sensing with the value of the information provided by observations in partially observable environments to enable efficient task execution in POMDPs.

The development of the VoI metric and the algorithms used to approximate it in this chapter represent a step towards autonomous decision-makers that are able to balance exploration and exploitation intelligently. The utility of such a metric is evident. In this thesis, we use VoI to construct macro-actions with guaranteed performance. VoI could also be used to determine when and how often agents should communicate in multi-agent settings or under communication constraints [161]. The generality of the approach—directly approximating the sensitivity of the POMDP value function to actions that reduce belief uncertainty—means that this formulation of VoI can be applied to any scientific decision-making problem, not only the dynamic tracking experiment explored here. Although the presentation in this chapter relied on a piecewise-linear and convex approximation of the POMDP value function in discrete problems, the general formulation would apply to any value-function approximation. Additionally, the analytical computation of VoI presented here could be used as the basis of a learning algorithm that generalizes the value of information across belief space more efficiently than direct approximation with point-based value function estimates. These are promising directions for future work that would enable the approaches presented here to be deployed in real-world oceanographic dynamic-tracking scenarios, such as those presented at the start of this chapter.

Chapter 5

Online Learning with Optimism and Delay

The exploration and exploitation trade-off manifests in different forms throughout decision-making problems. Chapter 3 and Chapter 4 work to balance exploration and exploitation in sequential decision-making problems, where an agent with a persistent state must use a model of the environment to accomplish some scientific task or objective under uncertainty. These problems were modeled as partially observable Markov decision processes, and task performance was used as the guide for trading off between exploration and exploitation when solving for a POMDP policy.

This chapter considers a new manifestation of the exploration and exploitation trade-off that arises in online learning. Online learning, sometimes called “learning with expert advice” is a paradigm in which a learner is pitted against a potentially adversarial environment [129, 152] and must learn from experience to make good decisions over time. In standard online learning formulations, the underlying environment is assumed to be adversarial and the learner does not maintain an explicit distribution over possible future outcomes. Unlike the POMDP formulations we saw in previous chapters, the decision-maker does not have a persistent state and the outcomes for each available action choice are fully observable at each timestep. Uncertainty enters the problem instead due to the unpredictable and potentially adversarial future behavior of the environment. The learner must strike a balance between trusting information provided in previous experiences by playing actions that have performed well in the past (exploitation) and hedging against potentially adversarial future outcomes by playing diverse action choices (exploration). Online learning seeks to strike a trade-off between exploration and exploitation that enables formal guarantees on the performance of the learner. Namely, we often want to guarantee that the learner will eventually learn to identify and play the optimal choice from among the candidate action set; these algorithms are known as “no-regret” algorithms.

The goal of this chapter is to develop online learning algorithms and corresponding performance bounds that can be used for real-world, scientific applications. We show how an online learner can optimally balance exploration and exploitation under several real-world constraints: when the feedback provided by the environment is delayed or provided incrementally; when the underlying environment has some predictability that should be leveraged; and when the learner must provide robust performance without relying on difficult-to-tune parameters. For each of these real-world challenges, we provide a novel, unified algorithmic framework and performance analysis that shows how online learners should adjust their exploration-exploitation balance to achieve asymptotically optimal performance. The motivating scientific application in the chapter is real-time, subseasonal temperature and precipitation forecasting, which is introduced in the following section.

Motivating Application: Model Selection for Subseasonal Forecasting. Improving our ability to forecast the weather and climate is of interest to all sectors of the economy and government agencies from the local to the national level. Weather forecasts 0-10 days ahead and climate forecasts seasons to decades ahead are currently used operationally in decision-making, and the accuracy and reliability of these forecasts has improved consistently in recent decades [180]. However, many critical applications – including water allocation, wildfire management, and drought and flood mitigation – require *subseasonal forecasts* (Fig. 5.1) with lead times between these two extremes, generally from 2 to 6 weeks in advance [117, 190].

While short-term forecasting accuracy is largely sustained by physics-based dynamical models, these deterministic methods have limited subseasonal accuracy due to the chaotic nature of atmospheric dynamics [107]. Indeed, subseasonal forecasting has long been considered a “predictability desert” due to its complex dependence on both local weather and global climate variables [183]. Recent large-scale research efforts have advanced the subseasonal capabilities of operational physics-based models [92, 136, 182], while parallel efforts have demonstrated the value of machine learning and deep learning methods in improving subseasonal forecasting [5, 31, 60, 68, 98, 166, 184, 187, 189, 194].

In real-time forecasting operations, we can play an ensemble of these state-of-the-art subseasonal models to achieve improved forecast accuracy, selecting the models that, for a given time and forecast horizon, minimize a chosen loss criteria. This requires striking a balance between choosing models that have performed well in the past and monitoring

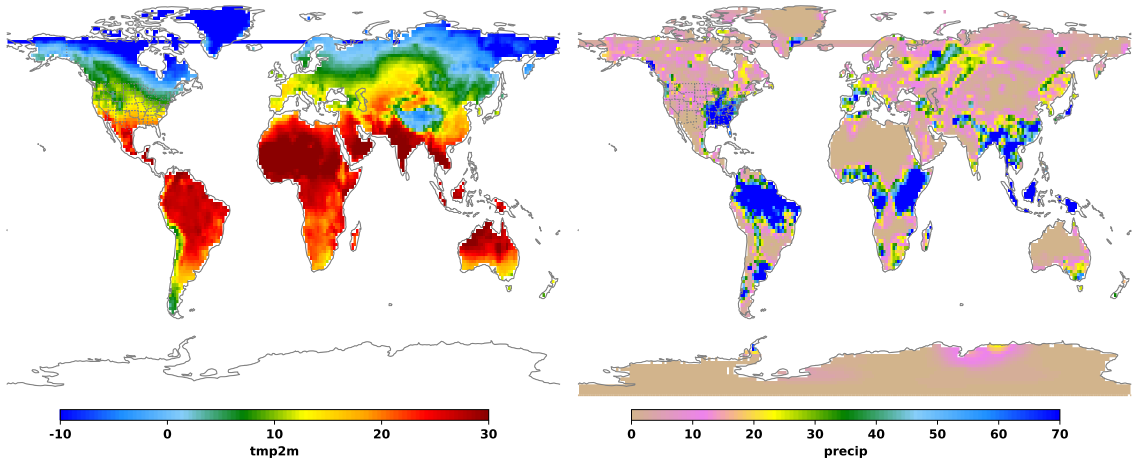


Figure 5.1: ***Subseasonal Temperature and Precipitation:*** *Examples of global temperature and precipitation outcomes on the subseasonal (2-6 weeks) timescale from Mouatadid et al. [123].*

low-performing models to ensure that they are leveraged if their performance improves. Model performance may be highly variable: some models may perform best in winter, others during a specific phase of the Madden Julien oscillation. Learning to produce a high-performing ensemble of models requires real-time, adaptive decision making and can be addressed with online learning methods.

Overview. Inspired by the demands of real-time climate and weather forecasting, the remainder of this chapter develops optimistic online learning algorithms that require no parameter tuning and have optimal regret guarantees under delayed feedback. The algorithms—DORM, DORM+, and AdaHedgeD—arise from a novel reduction of delayed online learning to optimistic online learning that reveals how optimistic hints can mitigate the regret penalty caused by delay. We pair this delay-as-optimism perspective with a new analysis of optimistic learning that exposes its robustness to hinting errors and a new meta-algorithm for learning effective hinting strategies in the presence of delay. We conclude by benchmarking our algorithms on four subseasonal climate forecasting tasks, demonstrating low regret relative to state-of-the-art forecasting models. This chapter is based on work appearing in Flaspohler et al. [47].

5.1 Online Learning with Optimism and Delay

Online learning is a sequential decision-making paradigm in which a learner is pitted against a potentially adversarial environment [129, 152]. At time t, the learner must select a play $\boldsymbol{w}_t$ from some set of possible plays $\mathbb{W}$. The environment then reveals the loss function ℓ_t and the learner pays the cost $\ell_t(\boldsymbol{w}_t)$. The learner uses information collected in previous rounds to improve its plays in subsequent rounds. *Optimistic* online learners additionally attempt to leverage any underlying predictability in the potentially adversarial environment by making use of side-information or "hints" about expected future losses to improve their plays. Over a period of length T, the goal of the learner is to minimize *regret*, an objective that quantifies the performance gap between the learner and the best possible constant play in retrospect in some competitor set $\mathbb{U}$:

$$\text{Regret}_T = \sup_{\boldsymbol{u} \in \mathbb{U}} \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}). \tag{5.1}$$

Adversarial online learning algorithms provide robust performance in many complex real-world online prediction problems such as climate or weather forecasting.

In traditional online learning paradigms, the loss for round t is revealed to the learner immediately at the end of round t. However, many real-world applications produce delayed feedback, i.e., the loss for round t is not available until round $t+D$ for some delay period D.[1] Existing delayed online learning algorithms achieve optimal worst-case regret rates against adversarial loss sequences, but each has drawbacks when deployed for real applications with short horizons T. Some use only a small fraction of the data to train each learner [74, 188]; others tune their parameters using uniform bounds on future gradients that are often challenging to obtain or overly conservative in applications [66, 75, 86, 114, 144]. Only the concurrent work of Hsieh et al. [66, Thm. 13] can make use of optimistic hints and only for the special case of unconstrained online gradient descent.

In this work, we aim to develop robust and practical algorithms for real-world delayed online learning. To this end, we introduce three novel algorithms—DORM, DORM+, and AdaHedgeD—that use every observation to train the learner, have no parameters to tune, exhibit optimal worst-case regret rates under delay, *and* enjoy improved performance when accurate hints for unobserved losses are available. We begin by formulating delayed online

[1] Our initial presentation will assume constant delay D, but we provide extensions to variable and unbounded delays in App. C.14.

learning as a special case of optimistic online learning and use this “delay-as-optimism” perspective to develop:

1. A formal reduction of delayed online learning to optimistic online learning (Lems. 5.4.1 and 5.4.2),
2. The first optimistic tuning-free and self-tuning algorithms with optimal regret guarantees under delay (DORM, DORM+, and AdaHedgeD),
3. A tightening of standard optimistic online learning regret bounds that reveals the robustness of optimistic algorithms to inaccurate hints (Thms. 5.4.1 and 5.4.2),
4. The first general analysis of follow-the-regularized-leader (Thms. 5.4.3 and 5.7.1) and online mirror descent algorithms (Thm. 5.4.4) with optimism and delay, and
5. The first meta-algorithm for learning a low-regret optimism strategy under delay (Thm. 5.8.2).

We apply our algorithms on the problem of subseasonal forecasting in Sec. 5.9. Subseasonal forecasting—predicting precipitation and temperature 2-6 weeks in advance—is a crucial task for allocating water resources and preparing for weather extremes [190]. Model selection for subseasonal forecasting can be posed as an online learning problem by allowing the play vector $\boldsymbol{w}_t$ to specify a convex ensemble of d different input forecasts from different models. The set of possible plays $\mathbb{W}$ is then the $(d-1)$-dimensional simplex, $\mathbb{W} = \triangle_{d-1}$. We aim to control regret against a competitor set $\mathbb{U}$ that consists of the d basis vectors, $\mathbb{U} = \{\mathbf{e}_i\}_{i=0}^{d-1}$, each representing the forecast of one of the d input models. Controlling regret against set $\mathbb{U}$ corresponds to playing an ensemble forecast that is at least as good as the best input model, an important objective for a real-time forecasting algorithm.

Subseasonal forecasting presents several challenges for online learning algorithms. First, real-time subseasonal forecasting suffers from delayed feedback: multiple forecasts are issued before receiving feedback on the first. Second, the regret horizons are short: a common evaluation period for semimonthly forecasting is one year, resulting in 26 total forecasts. Third, forecasters cannot have difficult-to-tune parameters in real-time, practical deployments. We demonstrate that our algorithms DORM, DORM+, and AdaHedgeD successfully overcome these challenges and achieve consistently low regret compared to the best forecasting models.

Notation Overview For integers a, b, we use the shorthand $[b] \triangleq \{1, \dots, b\}$ and $\boldsymbol{g}_{a:b} \triangleq \sum_{i=a}^{b} \boldsymbol{g}_i$. We say a function f is *proper* if it is somewhere finite and never $-\infty$.

We let $\partial f(\boldsymbol{w}) = \{\boldsymbol{g} \in \mathbb{R}^d : f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + \langle \boldsymbol{g}, \boldsymbol{u} - \boldsymbol{w} \rangle,\ \forall \boldsymbol{u} \in \mathbb{R}^d\}$ denote the set of *subgradients* of f at $\boldsymbol{w} \in \mathbb{R}^d$ and say f is *μ-strongly convex* over a convex set $\mathbb{W} \subseteq \operatorname{int}\operatorname{dom} f$ with respect to $\|\cdot\|$ with dual norm $\|\cdot\|_*$ if $\forall \boldsymbol{w}, \boldsymbol{u} \in \mathbb{W}$ and $\boldsymbol{g} \in \partial f(\boldsymbol{w})$, we have $f(\boldsymbol{u}) \geq f(\boldsymbol{w}) + \langle \boldsymbol{g}, \boldsymbol{u} - \boldsymbol{w} \rangle + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{u}\|^2$. For differentiable ψ, we define the Bregman divergence $\mathcal{B}_\psi(\boldsymbol{w}, \boldsymbol{u}) \triangleq \psi(\boldsymbol{w}) - \psi(\boldsymbol{u}) - \langle \nabla\psi(\boldsymbol{u}), \boldsymbol{w} - \boldsymbol{u} \rangle$. We define $\operatorname{diam}(\mathbb{W}) = \inf_{\boldsymbol{w}, \boldsymbol{w}' \in \mathbb{W}} \|\boldsymbol{w} - \boldsymbol{w}'\|$, $(r)_+ \triangleq \max(r, 0)$, and $\min(r, s)_+ \triangleq (\min(r, s))_+$.

5.2 Preliminaries: Standard and Optimistic Online Learning

The standard online learning algorithms, follow the regularized leader (FTRL) and online mirror descent (OMD), were introduced in Sec. 2.2. Although these classical algorithms provide optimal performance in the worst case, there is opportunity to improve both in several regards. First, as mentioned previously, these algorithm rely on a difficult-to-tune parameter λ that balances exploration and exploitation; each algorithm only achieves optimal regret when λ is tuned appropriately. Second, many loss sequences encountered in applications are not truly adversarial; real-world environments may produce losses that are partially predictable based on past experience and we would like to exploit that predictability to improve algorithm performance and provide tighter regret guarantees.

Optimistic online learning algorithms aim to improve performance when loss sequences are partially predictable, while remaining robust to adversarial sequences [see, e.g., 10, 28, 147, 169]. In optimistic online learning, the learner is provided with a "hint" in the form of a pseudo-loss $\tilde{\ell}_t$ at the start of round t that represents a guess for the true unknown future loss. The online learner can incorporate this hint before making play $\boldsymbol{w}_t$. In standard formulations of optimistic online learning, the convex pseudo-loss $\tilde{\ell}_t(\boldsymbol{w}_t)$ is added to the standard FTRL or OMD regularized objective function and leads to optimistic variants of these algorithms: optimistic FTRL [OFTRL, 146] and single-step optimistic OMD [SOOMD, 76, Sec. 7.2]. Let $\tilde{\boldsymbol{g}}_t \in \partial\tilde{\ell}_t(\boldsymbol{w}_{t-1})$ and $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{w}_t)$ denote subgradients of the pseudo-loss and true loss respectively. The inclusion of an optimistic hint leads to the following linearized update rules for play $\boldsymbol{w}_{t+1}$:

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_{1:t} + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{w} \rangle + \lambda\psi(\boldsymbol{w}), \tag{OFTRL}$$

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{w} \rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_t) \\ &\quad \text{with} \quad \tilde{\boldsymbol{g}}_0 = \boldsymbol{0} \quad \text{and arbitrary} \quad \boldsymbol{w}_0 \end{aligned} \tag{SOOMD}$$

where $\tilde{\boldsymbol{g}}_{t+1} \in \mathbb{R}^d$ is the hint subgradient, $\lambda \geq 0$ is a regularization parameter, and ψ is proper regularization function that is 1-strongly convex with respect to a norm $\|\cdot\|$. The optimistic learner enjoys reduced regret whenever the error in the provided hint $\|\boldsymbol{g}_{t+1} - \tilde{\boldsymbol{g}}_{t+1}\|_*$ is small [76, 146]. Common choices of optimistic hints include the last observed subgradient or average of previously observed subgradients [146], but the hint could also be produced by some auxiliary learning procedure that attempts to leverage seasonality or auxiliary information to predict future losses. We note that the standard FTRL and OMD updates can be recovered by setting the optimistic hints to zero.

5.3 Online Learning with Optimism and Delay: Algorithms

In the delayed feedback setting with constant delay of length D, the learner only observes $(\ell_i)_{i=1}^{t-D}$ before making play $\boldsymbol{w}_{t+1}$. In this setting, we propose counterparts of the OFTRL and SOOMD online learning algorithms, which we call *optimistic delayed FTRL (*ODFTRL*)* and *delayed optimistic online mirror descent (*DOOMD*)* respectively:

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{g}_{1:t-D} + \boldsymbol{h}_{t+1}, \boldsymbol{w}\rangle + \lambda\psi(\boldsymbol{w}) \tag{ODFTRL}$$

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{g}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t, \boldsymbol{w}\rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_t) \\ &\text{with} \quad \boldsymbol{h}_0 \triangleq \boldsymbol{0} \quad \text{and arbitrary} \quad \boldsymbol{w}_0, \end{aligned} \tag{DOOMD}$$

for hint vector $\boldsymbol{h}_{t+1}$. Our use of the notation $\boldsymbol{h}_{t+1}$ instead of $\tilde{\boldsymbol{g}}_{t+1}$ for the optimistic hint here is suggestive. Our regret analysis in Thms. 5.4.3 and 5.4.4 reveals that, instead of hinting only for the "future" missing loss $\boldsymbol{g}_{t+1}$, delayed online learners should uses hints $\boldsymbol{h}_t$ that guess at the summed subgradients of all delayed and future losses: $\boldsymbol{h}_t = \sum_{s=t-D}^{t} \tilde{\boldsymbol{g}}_s$.

5.4 Online Learning with Optimism and Delay: Regret Bounds

While it is easy to define new online learning algorithms, such as ODFTRL and DOOMD, it is often much more challenging to control the regret of the plays made by these algorithms. Controlling an algorithm's regret requires analyzing how the algorithm trades off between exploiting previously observed losses to make targeted plays and hedging against future, adversarial losses by making conservative plays.

To analyze the regret of the delayed ODFTRL and DOOMD algorithms, we make use of the first key insight of this chapter, which reduces online learning with delay to optimistic online learning, connecting these two branches of online learning for the first time: *Learning with delay is a special case of learning with optimism.* In particular, ODFTRL and DOOMD are instances of OFTRL and SOOMD respectively with a particularly "bad" choice of optimistic hint $\tilde{\boldsymbol{g}}_{t+1}$ that deletes the unobserved loss subgradients $\boldsymbol{g}_{t-D+1:t}$.

Lemma 5.4.1 (ODFTRL is OFTRL with a bad hint)**.** ODFTRL *is* OFTRL *with* $\tilde{\boldsymbol{g}}_{t+1} = \boldsymbol{h}_{t+1} - \sum_{s=t-D+1}^{t} \boldsymbol{g}_s$.

Lemma 5.4.2 (DOOMD is SOOMD with a bad hint)**.** DOOMD *is* SOOMD *with* $\tilde{\boldsymbol{g}}_{t+1} = \tilde{\boldsymbol{g}}_t + \boldsymbol{g}_{t-D} - \boldsymbol{g}_t + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t = \boldsymbol{h}_{t+1} - \sum_{s=t-D+1}^{t} \boldsymbol{g}_s$.

The implication of this reduction of delayed online learning to optimistic online learning is that *any* regret bound shown for undelayed OFTRL or SOOMD immediately yields a regret bound for ODFTRL and DOOMD under delay. As we demonstrate in the remainder of the chapter, this novel connection between delayed and optimistic online learning allows us to bound the regret of optimistic, self-tuning, and tuning-free algorithms for the first time under delay.

It is worth reflecting on the key property of OFTRL and SOOMD that enables the delay-to-optimism reduction: each algorithm depends on $\boldsymbol{g}_t$ and $\tilde{\boldsymbol{g}}_{t+1}$ only through the sum $\boldsymbol{g}_{1:t} + \tilde{\boldsymbol{g}}_{t+1}$.[2] For the "bad" hints of Lems. 5.4.1 and 5.4.2, these sums are observable even though $\boldsymbol{g}_t$ and $\tilde{\boldsymbol{g}}_{t+1}$ are not separately observable at time t due to delay. A number of alternatives to SOOMD have been proposed for optimistic OMD [28, 80, 146, 147]. Unlike SOOMD, these procedures all incorporate optimism in two steps, as in the updates

$$\boldsymbol{w}_{t+1/2} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_{t-1/2}) \quad \text{and} \tag{5.2}$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{w} \rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_{t+1/2}) \tag{5.3}$$

described in Rakhlin and Sridharan [146, Sec. 2.2]. It is unclear how to reduce delayed OMD to an instance of one of these two-step procedures, as knowledge of the unobserved $\boldsymbol{g}_t$ is needed to carry out the first step.

5.4.1 Delayed and Optimistic Regret Bounds

To demonstrate the utility of our delay-as-optimism perspective, we first present the following new regret bounds for OFTRL and SOOMD, proved in Apps. C.2 and C.3 respectively.

[2] For SOOMD, $\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t = \boldsymbol{g}_{1:t} + \tilde{\boldsymbol{g}}_{t+1} - (\boldsymbol{g}_{1:t-1} + \tilde{\boldsymbol{g}}_t)$.

Theorem 5.4.1 (OFTRL regret). *If ψ is nonnegative, then, for all $\boldsymbol{u} \in \mathbb{W}$, the* OFTRL *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda\psi(\boldsymbol{u}) + \frac{1}{\lambda}\sum_{t=1}^{T} \text{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*). \tag{5.4}$$

Theorem 5.4.2 (SOOMD regret). *If ψ is differentiable and $\tilde{\boldsymbol{g}}_{T+1} \triangleq \boldsymbol{0}$, then, $\forall \boldsymbol{u} \in \mathbb{W}$, the* SOOMD *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \mathcal{B}_{\lambda\psi}(\boldsymbol{u}, \boldsymbol{w}_0) + \tag{5.5}$$

$$\frac{1}{\lambda}\sum_{t=1}^{T} \text{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t\|_*). \tag{5.6}$$

Both results feature the robust Huber penalty [67]

$$\text{huber}(x, y) \triangleq \frac{1}{2}x^2 - \frac{1}{2}(|x| - |y|)_+^2 \leq \min(\frac{1}{2}x^2, |y||x|) \tag{5.7}$$

in place of the more common squared error term $\frac{1}{2}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*^2$. As a result, Thms. 5.4.1 and 5.4.2 strictly improve the rate-optimal OFTRL and SOOMD regret bounds of Mohri and Yang [119], Orabona [129], Rakhlin and Sridharan [146, Thm. 7.28] and Joulani et al. [76, Sec. 7.2] by revealing a previously undocumented robustness to inaccurate hints $\tilde{\boldsymbol{g}}_t$. We will use this robustness to large hint error $\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*$ to establish optimal regret bounds under delay.

As an immediate consequence of this regret analysis and our delay-as-optimism perspective, we obtain the first general analyses of FTRL and OMD with optimism and delay.

Theorem 5.4.3 (ODFTRL regret). *If ψ is nonnegative, then, for all $\boldsymbol{u} \in \mathbb{W}$, the* ODFTRL *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda\psi(\boldsymbol{u}) + \frac{1}{\lambda}\sum_{t=1}^{T} \boldsymbol{b}_{t,F} \quad \textit{for} \tag{5.8}$$

$$\boldsymbol{b}_{t,F} \triangleq \text{huber}(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_t\|_*). \tag{5.9}$$

Theorem 5.4.4 (DOOMD regret)**.** *If ψ is differentiable and $\boldsymbol{h}_{T+1} \triangleq \boldsymbol{g}_{T-D+1:T}$, then, for all $\boldsymbol{u} \in \mathbb{W}$, the* DOOMD *iterates $\boldsymbol{w}_t$ satisfy*

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \mathcal{B}_{\lambda\psi}(\boldsymbol{u}, \boldsymbol{w}_0) + \frac{1}{\lambda}\sum_{t=1}^{T} \boldsymbol{b}_{t,O} \quad \textit{for} \tag{5.10}$$

$$\boldsymbol{b}_{t,O} \triangleq \mathrm{huber}(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_*). \tag{5.11}$$

Our results show a compounding of regret due to delay: the $\boldsymbol{b}_{t,F}$ term of Thm. 5.4.3 is of size $\mathcal{O}(D+1)$ whenever $\|\boldsymbol{h}_t\|_* = \mathcal{O}(D+1)$, and the same holds for $\boldsymbol{b}_{t,O}$ of Thm. 5.4.4 if $\|\boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_* = \mathcal{O}(1)$. An optimal setting of λ therefore delivers $\mathcal{O}(\sqrt{(D+1)T})$ regret, yielding the minimax optimal rate for adversarial learning under delay [188]. Thms. 5.4.3 and 5.4.4 also reveal the heightened value of optimism in the presence of delay: in addition to providing an effective guess of the future subgradient $\boldsymbol{g}_t$, an optimistic hint can approximate the missing delayed feedback ($\sum_{s=t-D}^{t-1} \boldsymbol{g}_s$) and thereby significantly reduce the penalty of delay. If, on the other hand, the hints are a poor proxy for the missing loss subgradients, the novel huber term ensures that we still only pay the minimax optimal $\sqrt{D+1}$ penalty for delayed feedback.

Related work A classical approach to delayed feedback in online learning is the so-called "replication" strategy in which $D+1$ distinct learners take turns observing and responding to feedback [1, 74, 118, 188]. While minimax optimal in adversarial settings, this strategy has the disadvantage that each learner only sees $\frac{T}{D+1}$ losses and is completely isolated from the other replicates, exacerbating the problem of short prediction horizons. In contrast, we develop and analyze non-replicated delayed online learning strategies that use a combination of optimistic hinting and self-tuned regularization to mitigate the effects of delay while retaining optimal worst-case behavior.

We are not aware of prior analyses of DOOMD, and, to our knowledge, Thm. 5.4.3 and its adaptive generalization Thm. 5.7.1 provide the first general analysis of delayed FTRL, apart from the concurrent work of Hsieh et al. [66, Thm. 1]. Hsieh et al. [66, Thm. 13] and Quanrud and Khashabi [144, Thm. 2.1] focus only on delayed gradient descent, Korotin et al. [86] study General Hedging, and Joulani et al. [75, Thm. 4] and Quanrud and Khashabi [144, Thm. A.5] study non-optimistic OMD under delay. Thms. 5.4.3, 5.4.4, and 5.7.1 strengthen these results from the literature which feature a sum of subgradient norms ($\sum_{s=t-D}^{t-1} \|\boldsymbol{g}_s\|_*$ or $D\|\boldsymbol{g}_t\|_*$) in place of $\|\boldsymbol{h}_t - \sum_{s=t-D}^{t-1} \boldsymbol{g}_s\|_*$. Even in the absence of optimism, the latter can be significantly smaller: e.g., if the gradients $\boldsymbol{g}_s$ are i.i.d. mean-zero vectors, the former has size $\Omega(D)$ while the latter has expectation $\mathcal{O}(\sqrt{D})$. In the absence of optimism, McMahan and Streeter [114] obtain a bound comparable to Thm. 5.4.3 for the special case of one-dimensional unconstrained online gradient descent.

In the absence of delay, Cutkosky [36] introduces meta-algorithms for imbuing learning procedures with optimism while remaining robust to inaccurate hints; however, unlike OFTRL and SOOMD, the procedures of Cutkosky [36] require separate observation of $\tilde{\boldsymbol{g}}_{t+1}$ and each $\boldsymbol{g}_t$, making them unsuitable for our delay-to-optimism reduction.

5.5 Online Learning with Optimism and Delay: Regularization Tuning

The online learning algorithms introduced so far all include a regularization parameter λ. In theory and in practice, these algorithms only achieve low regret if the regularization parameter λ is chosen appropriately. In standard FTRL, for example, one such setting that achieves optimal regret is $\lambda = \sqrt{\frac{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_*^2}{\sup_{\boldsymbol{u} \in \mathbb{U}} \psi(\boldsymbol{u})}}$. This choice, however, cannot be used in practice as it relies on knowledge of all future unobserved loss subgradients. To make use of online learning algorithms, the tuning parameter λ is often set using coarse upper bounds on, e.g., the maximum possible subgradient norm. However, these bounds are often very conservative and lead to poor real-world performance.

In the following sections, we introduce two strategies for tuning regularization with optimism and delay. Sec. 5.6 introduces the DORM and DORM+ algorithms, variants of ODFTRL and DOOMD that are *entirely tuning-free*. Sec. 5.7 introduces the AdaHedgeD algorithm, an adaptive variant of ODFTRL that is *self-tuning*; a sequence of regularization parameters λ_t are set automatically using new, tighter bounds on algorithm regret. All three algorithms achieve the minimax optimal regret rate under delay, support optimism, and have strong real-world performance as shown in Sec. 5.9.

5.6 Tuning-free Learning with Optimism and Delay

Regret matching (RM) [20, 54] and regret matching+ (RM+) [177] are online learning algorithms that have strong empirical performance. RM was developed to find correlated equilibria in two-player games and is commonly used to minimize regret over the simplex. RM+ is a modification of RM designed to accelerate convergence and used to effectively solve the game of Heads-up Limit Texas Hold'em poker [22]. RM and RM+ support neither optimistic hints nor delayed feedback, and known regret bounds have a suboptimal scaling with respect to the problem dimension d [26, 131]. To extend these algorithms to the delayed and optimistic setting and recover the optimal regret rate, we introduce our generalizations, *delayed optimistic regret matching* (DORM)

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \tilde{\boldsymbol{w}}_{t+1}/\langle \mathbf{1}, \tilde{\boldsymbol{w}}_{t+1}\rangle \quad \text{for} \\ \tilde{\boldsymbol{w}}_{t+1} &\triangleq \max(\mathbf{0}, (\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})/\lambda)^{q-1} \end{aligned} \tag{DORM}$$

and *delayed optimistic regret matching+* (DORM+)

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \tilde{\boldsymbol{w}}_{t+1}/\langle \mathbf{1}, \tilde{\boldsymbol{w}}_{t+1}\rangle \text{ for} \\ \boldsymbol{h}_0 &= \tilde{\boldsymbol{w}}_0 \triangleq \mathbf{0}, \\ \tilde{\boldsymbol{w}}_{t+1} &\triangleq \max\left(\mathbf{0}, \tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t)/\lambda\right)^{q-1}, \end{aligned} \tag{DORM+}$$

Each algorithm makes use of an instantaneous regret vector $\boldsymbol{r}_t \triangleq \mathbf{1}\langle \boldsymbol{g}_t, \boldsymbol{w}_t\rangle - \boldsymbol{g}_t$ that quantifies the relative performance of each expert with respect to the play $\boldsymbol{w}_t$ and the linearized loss subgradient $\boldsymbol{g}_t$. The updates also include a parameter $q \geq 2$ and its conjugate exponent $p = q/(q-1)$ that is set to recover the minimax optimal scaling of regret with the number of experts (see Cor. 5.6.1). We note that DORM and DORM+ recover the standard RM and RM+ algorithms when $D = 0$, $\lambda = 1$, $q = 2$, and $\boldsymbol{h}_t = \mathbf{0}$, $\forall t$.

5.6.1 Tuning-free Regret Bounds

To bound the regret of the DORM and DORM+ plays, we prove that DORM is an instance of ODFTRL and DORM+ is an instance of DOOMD. This connection enables us to immediately provide regret guarantees for these regret-matching algorithms under delayed feedback and with optimism. We first highlight a remarkable property of DORM and DORM+ that is the basis of their tuning-free nature. Under mild conditions, the normalized DORM and DORM+ iterates $\boldsymbol{w}_t$ are *independent* of the choice of regularization parameter λ.

Lemma 5.6.1 (DORM and DORM+ are independent of λ)**.** *If the subgradient $\boldsymbol{g}_t$ and hint $\boldsymbol{h}_{t+1}$ only depend on λ through $(\boldsymbol{w}_s, \lambda^{q-1}\tilde{\boldsymbol{w}}_s, \boldsymbol{g}_{s-1}, \boldsymbol{h}_s)_{s\leq t}$ and $(\boldsymbol{w}_s, \lambda^{q-1}\tilde{\boldsymbol{w}}_s, \boldsymbol{g}_s, \boldsymbol{h}_s)_{s\leq t}$ respectively, then the* DORM *and* DORM+ *iterates $(\boldsymbol{w}_t)_{t\geq 1}$ are independent of the choice of $\lambda > 0$.*

Lem. 5.6.1, proved in App. C.5, implies that DORM and DORM+ are *automatically* optimally tuned with respect to λ, even when run with a default value of $\lambda = 1$. Hence, these algorithms are tuning-free, a very appealing property for real-world deployments of online learning.

To show that DORM and DORM+ also achieve optimal regret scaling under delay, we connect them to ODFTRL and DOOMD operating on the nonnegative orthant with a special surrogate loss $\hat{\ell}_t$ (see App. C.4 for our proof):

Lemma 5.6.2 (DORM is ODFTRL and DORM+ is DOOMD)**.** *The* DORM *and* DORM+ *iterates are proportional to* ODFTRL *and* DOOMD *iterates respectively with* $\mathbb{W} \triangleq \mathbb{R}^d_+$, $\psi(\tilde{\boldsymbol{w}}) = \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_p^2$, *and loss* $\hat{\ell}_t(\tilde{\boldsymbol{w}}) = \langle \tilde{\boldsymbol{w}}, -\boldsymbol{r}_t \rangle$.

Lem. 5.6.2 enables the following optimally-tuned regret bounds for DORM and DORM+ run with any choice of λ:

Corollary 5.6.1 (DORM and DORM+ regret)**.** *Under the assumptions of Lem. 5.6.1, for all* $\boldsymbol{u} \in \triangle_{d-1}$ *and any choice of* $\lambda > 0$, *the* DORM *and* DORM+ *iterates* $\boldsymbol{w}_t$ *satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \le \inf_{\lambda > 0} \tfrac{\lambda}{2}\|\boldsymbol{u}\|_p^2 + \tfrac{1}{\lambda(p-1)} \textstyle\sum_{t=1}^T \boldsymbol{b}_{t,q} \tag{5.12}$$

$$= \sqrt{\tfrac{\|\boldsymbol{u}\|_p^2}{2(p-1)} \textstyle\sum_{t=1}^T \boldsymbol{b}_{t,q}} \le \sqrt{\tfrac{d^{2/q}(q-1)}{2} \textstyle\sum_{t=1}^T \boldsymbol{b}_{t,\infty}} \tag{5.13}$$

where $\boldsymbol{h}_{T+1} \triangleq \boldsymbol{r}_{T-D+1:T}$ *and, for each* $c \in [2, \infty]$,

$$\boldsymbol{b}_{t,c} \overset{(\text{DORM})}{=} \text{huber}(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_c, \|\boldsymbol{r}_t\|_c) \quad \textit{and} \tag{5.14}$$

$$\boldsymbol{b}_{t,c} \overset{(\text{DORM+})}{=} \text{huber}(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_c^2, \tag{5.15}$$

$$\|\boldsymbol{r}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_c). \tag{5.16}$$

If, in addition, $q = \arg\min_{q' \ge 2} d^{2/q'}(q'-1)$, *then* $\text{Regret}_T(\boldsymbol{u}) \le \sqrt{(2\log_2(d) - 1)\sum_{t=1}^T \boldsymbol{b}_{t,\infty}}$.

Cor. 5.6.1, proved in App. C.6, suggests a natural hinting strategy for reducing the regret of DORM and DORM+: predict the sum of unobserved instantaneous regrets $\sum_{s=t-D}^{t} \boldsymbol{r}_s$. We explore this strategy empirically in Sec. 5.9. Cor. 5.6.1 also highlights the value of the q parameter in DORM and DORM+: using the easily computed value

$q = \arg\min_{q' \geq 2} d^{2/q'}(q'-1)$ yields the minimax optimal $\sqrt{\log_2(d)}$ dependence of regret on dimension [26, 131]. By Lem. 5.6.2, setting q in this way is equivalent to selecting a robust $\frac{1}{2}\|\cdot\|_p^2$ regularizer [50] for the underlying ODFTRL and DOOMD problems.

Related work Without delay, Farina et al. [42] independently developed optimistic versions of RM and RM+ by reducing them to OFTRL and a two-step variant of optimistic OMD (5.3). Unlike SOOMD, this two-step optimistic OMD requires separate observation of $\tilde{\boldsymbol{g}}_{t+1}$ and $\boldsymbol{g}_t$, making it unsuitable for our delay-as-optimism reduction and resulting in a different algorithm from DORM+ even when $D = 0$. In addition, their regret bounds and prior bounds for RM and RM+ (special cases of DORM and DORM+ with $q = 2$) have suboptimal regret when the dimension d is large [22, 201].

5.7 Self-tuned Learning with Optimism and Delay

In this section, we analyze an adaptive version of ODFTRL with time-varying regularization $\lambda_t\psi$ and develop strategies for setting λ_t appropriately in the presence of optimism and delay. We begin with a new general regret analysis of optimistic delayed *adaptive* FTRL (ODAFTRL)

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{g}_{1:t-D} + \boldsymbol{h}_{t+1}, \boldsymbol{w}\rangle + \lambda_{t+1}\psi(\boldsymbol{w}) \tag{ODAFTRL}$$

where $\boldsymbol{h}_{t+1} \in \mathbb{R}^d$ is an arbitrary hint vector revealed before $\boldsymbol{w}_{t+1}$ is generated, ψ is 1-strongly convex with respect to a norm $\|\cdot\|$, and $\lambda_t \geq 0$ is a regularization parameter.

Theorem 5.7.1 (ODAFTRL regret). *If ψ is nonnegative and λ_t is non-decreasing in t, then, $\forall \boldsymbol{u} \in \mathbb{W}$, the* ODAFTRL *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda_T\psi(\boldsymbol{u}) + \sum_{t=1}^{T} \min\left(\frac{\boldsymbol{b}_{t,F}}{\lambda_t}, \boldsymbol{a}_{t,F}\right) \quad \textit{with} \tag{5.17}$$

$$\boldsymbol{b}_{t,F} \triangleq \text{huber}(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_t\|_*) \quad \textit{and} \tag{5.18}$$

$$\boldsymbol{a}_{t,F} \triangleq \text{diam}(\mathbb{W}) \min\left(\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_t\|_*\right). \tag{5.19}$$

The proof of this result in App. C.7 builds on a new regret bound for undelayed optimistic adaptive FTRL (OAFTRL). In the absence of delay ($D = 0$), Thm. 5.7.1 strictly improves existing regret bounds [76, 119, 146] for OAFTRL by providing tighter guarantees whenever the hinting error $\|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{g}_t\|_*$ is larger than the subgradient magnitude $\|\boldsymbol{g}_t\|_*$. In the presence of delay, Thm. 5.7.1 benefits both from robustness to hinting error in the worst case and the ability to exploit accurate hints in the best case. The bounded-domain factors $\boldsymbol{a}_{t,F}$ strengthen both standard OAFTRL regret bounds and the concurrent bound of Hsieh et al. [66, Thm. 1] when $\text{diam}(\mathbb{W})$ is small and will enable us to design practical λ_t-tuning strategies under delay without any prior knowledge of unobserved subgradients. We now turn to these self-tuning protocols.

5.7.1 Conservative Tuning with Delayed Upper Bound

We must select a λ_t sequence the leads to optimal regret bounds for the ODAFTRL algorithm. Setting aside the $\boldsymbol{a}_{t,F}$ bounded-domain factors in Thm. 5.7.1 for now, the adaptive sequence $\lambda_t = \sqrt{\frac{\sum_{s=1}^{t} \boldsymbol{b}_{s,F}}{\sup_{\boldsymbol{u}\in\mathbb{U}} \psi(\boldsymbol{u})}}$ is known to be a near-optimal minimizer of the ODAFTRL regret bound [115, Lemma 1]. However, this value is unobservable at time t. A common strategy is to play the conservative value $\lambda_t = \sqrt{\frac{(D+1)B_0 + \sum_{s=1}^{t-D-1} \boldsymbol{b}_{s,F}}{\sup_{\boldsymbol{u}\in\mathbb{U}} \psi(\boldsymbol{u})}}$, where B_0 is a uniform upper bound on the unobserved $\boldsymbol{b}_{s,F}$ terms [75, 114]. In practice, this requires computing an *a priori* upper bound on any subgradient norm that could possibly arise and often leads to extreme over-regularization (see Sec. 5.9).

As a preliminary step towards fully adaptive settings of λ_t, we analyze in App. C.8 a new *delayed upper bound* (DUB) tuning strategy which relies only on observed $\boldsymbol{b}_{s,F}$ terms and does not require upper bounds for future losses.

Theorem 5.7.2 (DUB regret)**.** *Fix $\alpha > 0$, and, for $\boldsymbol{a}_{t,F}, \boldsymbol{b}_{t,F}$ as in* (5.18)*, consider the* delayed upper bound *(*DUB*) sequence*

$$\lambda_{t+1} = \frac{2}{\alpha} \max_{j \le t-D-1} \boldsymbol{a}_{j-D+1:j,F} \tag{DUB}$$

$$+ \frac{1}{\alpha} \sqrt{\sum_{i=1}^{t-D} \boldsymbol{a}_{i,F}^2 + 2\alpha \boldsymbol{b}_{i,F}}. \tag{5.20}$$

If ψ is nonnegative, then, for all $\boldsymbol{u} \in \mathbb{W}$, the ODAFTRL *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \big(\frac{\psi(\boldsymbol{u})}{\alpha} + 1\big) \tag{5.21}$$

$$\Big(2 \max_{t\in[T]} \boldsymbol{a}_{t-D:t-1,F} + \sqrt{\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha \boldsymbol{b}_{t,F}}\Big). \tag{5.22}$$

As desired, the DUB setting of λ_t depends only on previously observed $\boldsymbol{a}_{t,F}$ and $\boldsymbol{b}_{t,F}$ terms and achieves optimal regret scaling with the delay period D. However, the terms $\boldsymbol{a}_{t,F}$, $\boldsymbol{b}_{t,F}$ are themselves potentially loose upper bounds for the instantaneous regret at time t. In the following section, we show how the DUB regularization setting can be refined further to produce AdaHedgeD adaptive regularization.

5.7.2 Refined Tuning with AdaHedgeD

As noted by de Rooij et al. [39], Erven et al. [40], Orabona [129], the effectiveness of an adaptive regularization setting λ_t that uses an upper bound on regret (such as $\boldsymbol{b}_{t,F}$) relies heavily on the tightness of that bound. In practice, we want to set λ_t using as tight a bound as possible. Our next result introduces a new tuning sequence that can be used with delayed feedback and is inspired by the popular AdaHedge algorithm [40]. It makes use of the tightened regret analysis underlying Thm. 5.7.1 to enable tighter settings of λ_t compared to DUB, while still controlling algorithm regret (see proof in App. C.9).

Theorem 5.7.3 (AdaHedgeD regret)**.** *Fix $\alpha > 0$, and consider the* delayed AdaHedge-style *(*AdaHedgeD*) sequence*

$$\lambda_{t+1} = \tfrac{1}{\alpha} \textstyle\sum_{s=1}^{t-D} \delta_s \qquad \textit{for} \tag{AdaHedgeD}$$

$$\delta_t \triangleq \min(F_{t+1}(\boldsymbol{w}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t), \quad \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \bar{\boldsymbol{w}}_t \rangle, \tag{5.23}$$

$$F_{t+1}(\hat{\boldsymbol{w}}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t \rangle)_+ \tag{5.24}$$

$$\textit{with} \quad \bar{\boldsymbol{w}}_t \triangleq \arg\min\nolimits_{\boldsymbol{w}\in\mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t), \tag{5.25}$$

$$\hat{\boldsymbol{w}}_t \triangleq \arg\min_{\boldsymbol{w}\in\mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t) + \tag{5.26}$$

$$\min(\frac{\|\boldsymbol{g}_t\|_*}{\|\boldsymbol{h}_t - \boldsymbol{g}_{t-D:t}\|_*}, 1) \langle \boldsymbol{h}_t - \boldsymbol{g}_{t-D:t}, \boldsymbol{w} \rangle, \tag{5.27}$$

$$\textit{and} \quad F_{t+1}(\boldsymbol{w}, \lambda_t) \triangleq \lambda_t \psi(\boldsymbol{w}) + \langle \boldsymbol{g}_{1:t}, \boldsymbol{w} \rangle. \tag{5.28}$$

If ψ is nonnegative, then, for all $\boldsymbol{u} \in \mathbb{W}$, the ODAFTRL *iterates satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \big(\frac{\psi(\boldsymbol{u})}{\alpha} + 1\big) \tag{5.29}$$

$$\Big(2 \max_{t \in [T]} \boldsymbol{a}_{t-D:t-1,F} + \sqrt{\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha \boldsymbol{b}_{t,F}}\Big). \tag{5.30}$$

Remarkably, Thm. 5.7.3 yields a minimax optimal $\mathcal{O}(\sqrt{(D+1)T} + D)$ dependence on the delay parameter and nearly matches the Thm. 5.4.3 regret of the optimal constant λ tuning. Although this regret bound is identical to that in Thm. 5.7.2, in practice the λ_t values produced by AdaHedgeD can be orders of magnitude smaller than those of DUB, granting additional adaptivity. We evaluate the practical implications of these λ_t settings in Sec. 5.9.

As a final note, when ψ is bounded on $\mathbb{U}$, we recommend choosing $\alpha = \sup_{\boldsymbol{u} \in \mathbb{U}} \psi(\boldsymbol{u})$ so that $\frac{\psi(\boldsymbol{u})}{\alpha} \leq 1$. For negative entropy regularization $\psi(\boldsymbol{u}) = \sum_{j=1}^{d} \boldsymbol{u}_j \ln(\boldsymbol{u}_j) + \ln(d)$ on the simplex $\mathbb{U} = \mathbb{W} = \triangle_{d-1}$, this yields $\alpha = \ln(d)$ and a regret bound with minimax optimal $\sqrt{\ln(d)}$ dependence on d [26, 131].

Related work Our AdaHedgeD δ_t terms differ from standard AdaHedge increments [see, e.g., 129, Sec. 7.6] due to the accommodation of delay, the incorporation of optimism, and the inclusion of the final two terms in the min. These non-standard terms are central to reducing the impact of delay on our regret bounds. Prior and concurrent approaches to adaptive tuning under delay do not incorporate optimism and require an explicit upper bound on all future subgradient norms, a quantity which is often difficult to obtain or very loose [66, 75, 114]. Our optimistic algorithms, DUB and AdaHedgeD, admit comparable regret guarantees (Thms. 5.7.2 and 5.7.3) but require no prior knowledge of future subgradients.

5.8 Learning to Hint with Delay

As we have seen, optimistic hints play an important role in online learning under delay: effective hinting can counteract the increase in regret under delay. In this section, we consider the problem of choosing amongst several competing hinting strategies. We show that this problem can again be treated as a delayed online learning problem. In the following, we will call the original online learning problem the "base problem" and the learning-to-hint problem the "hinting problem."

Suppose that, at time t, we observe the hints $\tilde{\boldsymbol{g}}_t$ of m different hinters arranged into a

$d \times m$ matrix $\mathbf{H}_t$. Each column of $\mathbf{H}_t$ is one hinter's best estimate of the sum of missing loss subgradients $\boldsymbol{g}_{t-D:t}$. Our aim is to output a sequence of combined hints $\boldsymbol{h}_t(\omega_t) \triangleq \mathbf{H}_t\omega_t$ with low regret relative to the best constant combination strategy $\omega \in \Omega \triangleq \triangle_{m-1}$ in hindsight. To achieve this using delayed online learning, we make use of a convex loss function $l_t(\omega)$ for the hint learner that upper bounds the base learner regret.

Assumption 5.8.1 (Convex regret bound)**.** *For any hint sequence $(\boldsymbol{h}_t)_{t=1}^T$ and $\boldsymbol{u} \in \Omega$, the base problem admits the regret bound $\mathrm{Regret}_T(\boldsymbol{u}) \leq C_0(\boldsymbol{u}) + C_1(\boldsymbol{u})\sqrt{\sum_{t=1}^T f_t(\boldsymbol{h}_t)}$ for $C_1(\boldsymbol{u}) \geq 0$ and convex functions f_t independent of $\boldsymbol{u}$.*

As we detail in App. C.11, Assump. 5.8.1 holds for all of the learning algorithms introduced in this thesis. For example, by Cor. 5.6.1, if the base learner is DORM, we may choose $C_0(\boldsymbol{u}) = 0$, $C_1(\boldsymbol{u}) = \sqrt{\frac{\|\boldsymbol{u}\|_p^2}{2(p-1)}}$, and the $\mathcal{O}(D+1)$ convex function $f_t(\boldsymbol{h}_t) = \|\boldsymbol{r}_t\|_q \|\boldsymbol{h}_t - \sum_{s=t-D}^t \boldsymbol{r}_s\|_q \geq \boldsymbol{b}_{t,q}$.[3]

For any base learner satisfying Assump. 5.8.1, we choose $l_t(\omega) = f_t(\mathbf{H}_t\omega)$ as our hinting loss, use the tuning-free DORM+ algorithm to output the combination weights ω_t on each round, and provide the hint $\boldsymbol{h}_t(\omega_t) = \mathbf{H}_t\omega_t$ to the base learner. The following result, proved in App. C.10, shows that this learning to hint strategy performs nearly as well as the best constant hint combination strategy in retrospect.

Theorem 5.8.2 (Learning to hint regret)**.** *Suppose the base problem satisfies Assump. 5.8.1 and the hinting problem is solved with* DORM+ *hint iterates ω_t, hinting losses $l_t(\omega) = f_t(\mathbf{H}_t\omega)$, no meta-hints for the hinting problem, and $q = \arg\min_{q' \geq 2} m^{2/q'}(q'-1)$. Then the base problem with hints $\boldsymbol{h}_t(\omega_t) = \mathbf{H}_t\omega_t$ satisfies*

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq C_0(\boldsymbol{u}) + C_1(\boldsymbol{u})\sqrt{\inf_{\omega\in\Omega} \sum_{t=1}^T f_t(\boldsymbol{h}_t(\omega))} \tag{5.31}$$

$$+ C_1(\boldsymbol{u})\big((2\log_2(m) - 1)(\frac{1}{2}\xi_T + \sum_{t=1}^{T-1} \mathrm{huber}(\xi_t, \zeta_t))\big)^{1/4} \tag{5.32}$$

$$\textit{for} \quad \xi_t \triangleq 4(D+1) \sum_{s=t-D}^t \|\gamma_s\|_\infty^2, \quad \gamma_t \in \partial l_t(\omega_t), \tag{5.33}$$

$$\textit{and} \quad \zeta_t \triangleq 4\|\gamma_{t-D}\|_\infty \sum_{s=t-D}^t \|\gamma_s\|_\infty. \tag{5.34}$$

[3] The alternative choice $f_t(\boldsymbol{h}_t) = \frac{1}{2}\|\boldsymbol{h}_t - \sum_{s=t-D}^t \boldsymbol{r}_s\|_q^2$ also bounds regret but may have size $\Theta((D+1)^2)$.

To quantify the size of this regret bound, consider again the DORM base learner with $f_t(\boldsymbol{h}_t) = \|\boldsymbol{r}_t\|_q \|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_q$. By Lem. C.11.2 in App. C.11, $\|\gamma_t\|_\infty \leq d^{1/q} \|\mathbf{H}_t\|_\infty \|\boldsymbol{r}_t\|_q$ for $\|\mathbf{H}_t\|_\infty$ the maximum absolute entry of $\mathbf{H}_t$. Each column of $\mathbf{H}_t$ is a sum $D+1$ subgradient hints, so $\|\mathbf{H}_t\|_\infty$ is $\mathcal{O}(D+1)$. Thus, for this choice of hinter loss, the huber(ξ_t, ζ_t) term is $\mathcal{O}((D+1)^3)$, and the hint learner suffers only $\mathcal{O}(T^{1/4}(D+1)^{3/4})$ additional regret from learning to hint. Notably, this additive regret penalty is $\mathcal{O}(\sqrt{(D+1)T})$ if $D = \mathcal{O}(T)$ (and $o(\sqrt{(D+1)T})$ when $D = o(T)$), so the learning to hint strategy of Thm. 5.8.2 preserves minimax optimal regret rates.

Related work Rakhlin and Sridharan [146, Sec. 4.1] propose and analyze a method to learn optimism strategies for a two-step OMD base learner. Unlike Thm. 5.8.2, the approach does not accommodate delay, and the analyzed regret is only with respect to single hinting strategies $\omega \in \{\mathbf{e}_j\}_{j \in [m]}$ rather than combination strategies, $\omega \in \triangle_{m-1}$.

5.9 Experiments

We apply the online learning techniques developed in this thesis to the problem of adaptive ensembling for subseasonal weather forecasting. Subseasonal forecasting is the problem predicting meteorological variables, often temperature and precipitation, 2-6 weeks in advance. These mid-range forecasts are critical for managing water resources and mitigating wildfires, droughts, floods, and other extreme weather events [68]. However, the subseasonal forecasting task is notoriously difficult due to the joint influences of short-term initial conditions and long-term boundary conditions [190].

To improve subseasonal weather forecasting capabilities, the US Department of Reclamation launched the Sub-Seasonal Climate Forecast Rodeo competition [128], a yearlong real-time forecasting competition for the Western United States. Our experiments are based on Flaspohler et al. [48], a snapshot of public subseasonal model forecasts including both physics-based and machine learning models. These models were developed for the subseasonal forecasting challenge and make semimonthly forecasts for the contest period (19 October 2019 – 29 September 2020).

To expand our evaluation beyond the subseasonal forecasting competition, we used the forecasts in Flaspohler et al. [48] for analogous yearlong periods (26 semi-monthly dates starting from the last Wednesday in October) beginning in Oct. 2010 and ending in Sep. 2020. Throughout, we refer to the yearlong period beginning in Oct. 2010 – Sep. 2011 as the 2011 year and so on for each subsequent year. For each forecast date t, the models in Flaspohler et al. [48] were trained only on data available at time t and model hyper-parameters were tuned to optimize average RMSE loss on the 3-year period preceding the forecast date t. For a few of the forecast dates, one or more models had missing forecasts; only dates for which all models have forecasts were used in evaluation.

5.9.1 Problem Definition

Denote the set of $d = 6$ input models $\{\mathcal{M}_1, \dots \mathcal{M}_d\}$ with labels: `llr` (Model1), `multillr` (Model2), `tuned_catboost` (Model3), `tuned_cfsv2` (Model4), `tuned_doy` (Model5) and `tuned_salient_fri` (Model6). On each semimonthly forecast date, each model $\mathcal{M}_i$ makes a prediction for each of two meteorological variables (cumulative precipitation and average temperature over 14 days) and two forecasting horizons (3-4 weeks and 5-6 weeks). For the 3-4 week and 5-6 horizons respectively, the forecaster experiences a delay of $D = 2$ and $D = 3$ forecasts. Each model makes a total of $T = 26$ semimonthly forecasts for these four tasks.

At each time t, each input model $\mathcal{M}_i$ produces a prediction at $G = 514$ gridpoints in the Western United States: $\boldsymbol{x}_{t,i}^c \in \mathbb{R}^G = \mathcal{M}_i(t)$ for task c at time t. Let $\mathbf{X}_t^c \in \mathbb{R}^{G \times d}$ be the matrix containing each input model's predictions as columns. The true meteorological outcome for task c is $\boldsymbol{y}_t^c \in \mathbb{R}^G$. As online learning is performed for each task separately, we drop the task superscript c in the following.

At each timestep, the online learner makes a forecast prediction $\hat{\boldsymbol{y}}_t$ by playing $\boldsymbol{w}_t \in \mathbb{W} = \triangle_{d-1}$, corresponding to a convex combination of the individual models: $\hat{\boldsymbol{y}}_t = \mathbf{X}_t \boldsymbol{w}_t$. The learner then incurs a loss for the play $\boldsymbol{w}_t$ according to the root mean squared (RMSE) error over the geography of interest:

$$\ell_t(\boldsymbol{w}_t) = \frac{1}{\sqrt{G}} \|\boldsymbol{y}_t - \mathbf{X}_t \boldsymbol{w}_t\|_2, \tag{5.35}$$

$$\partial \ell_t(\boldsymbol{w}_t) \ni \boldsymbol{g}_t = \begin{cases} \frac{\mathbf{X}_t^\top (\mathbf{X}_t \boldsymbol{w}_t - \boldsymbol{y}_t)}{\sqrt{G} \|\mathbf{X}_t \boldsymbol{w}_t - \boldsymbol{y}_t\|_2} & \text{if} \quad \mathbf{X}_t \boldsymbol{w}_t - \boldsymbol{y}_t \neq \mathbf{0} \\ \mathbf{0} & \text{if} \quad \mathbf{X}_t \boldsymbol{w}_t - \boldsymbol{y}_t = \mathbf{0} \end{cases} \tag{5.36}$$

Our objective for the subseasonal forecasting application is to produce an adaptive ensemble forecast that competes with the best input model over the yearlong period. Hence, in our evaluation, we take the competitor set to be the set of individual models $\mathbb{U} = \{\mathbf{e}_i : i \in [d]\}$.

We evaluate the relative merits of the delayed online learning techniques presented by computing yearly regret and mean RMSE for the ensemble plays made by the online leaner in each year from 2011-2020. Unless otherwise specified, all online learning algorithms use the `recent_g` hint $\tilde{\boldsymbol{g}}_s$, which approximates each unobserved subgradient at time t with the most recent observed subgradient $\boldsymbol{g}_{t-D-1}$.

See App. C.13 for algorithmic details and App. C.12 for extended experimental results. A Python library for Optimistic Online Learning under Delay (PoolD) and experiment code are available at https://github.com/geflaspohler/poold.

5.9.2 Results

Competing with the best input model The primary benefit of online learning in this setting is its ability to achieve small average regret, i.e., to perform nearly as well as the best input model in the competitor set $\mathbb{U}$ without knowing which is best in advance. We run our three delayed online learners—DORM, DORM+, and AdaHedgeD—on all four subseasonal prediction tasks and measure their average loss.

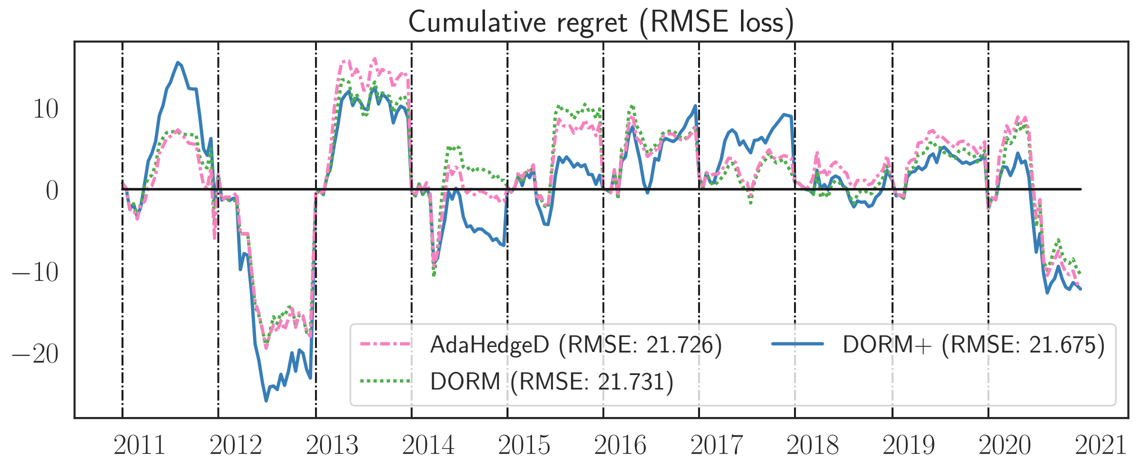


*Figure 5.2: **Overall Performance**: Yearly cumulative regret under RMSE loss for the the Precip. 3-4w task. The zero line corresponds to the performance of the best input model in a given year. Cumulative regret is reset to zero at the start of each year-long period (denoted with a dashed line) and accumulates throughout the year. At the end of the year period, online learners with negative cumulative regret values have successfully outperformed the best input model. The more negative the value, the better the performance of the online learner.*

The average yearly RMSE for the three online learning algorithms and the six input models is shown in Table 5.1. The DORM+ algorithm tracks the performance of the best input model for all tasks except Temp. 5-6w. All online learning algorithms achieve negative regret for both precipitation tasks. Fig. 5.2 shows the yearly cumulative regret (in terms of the RMSE loss) of the online learning algorithms over the 10-year evaluation period. There are several years (e.g., 2012, 2014, 2020) in which all online learning algorithms substantially outperform the best input forecasting model (at the end of a year-long period, the cumulative regret for a learner is negative). The consistently low regret year-to-year of DORM+ compared to DORM and AdaHedgeD makes it a promising candidate for real-world delayed subseasonal forecasting. Notably, RM+ (a special case of DORM+) is known to have small *tracking regret*, i.e., it competes well even with strategies that switch between input models a bounded number of times [177, Thm. 2]. We suspect that this is one source of DORM+'s superior performance. We also note that the self-tuned AdaHedgeD performs comparably to the optimally-tuned DORM, demonstrating the effectiveness of our self-tuning strategy.

Table 5.1: ***Average RMSE of the 2011-2020 Semimonthly Forecasts****: The average RMSE for online learning algorithms (left) and input models (right) over a 10-year evaluation period with the top-performing learners bolded and blue. In each task, the online learners compare favorably with the best input model and avoid the performance of the lower-performing models by leaning to downweigh them.*

	AdaHedgeD	DORM	DORM+	Best Input Model	Worst Model
Precip. 3-4w	21.726	21.731	**21.675**	21.973	23.344
Precip. 5-6w	21.868	21.957	**21.838**	21.993	23.257
Temp. 3-4w	2.273	2.259	**2.247**	2.253	2.508
Temp. 5-6w	2.316	2.316	**2.303**	2.270	2.569

Impact of regularization We evaluate the impact of the three regularization strategies developed in this thesis: 1) the upper bound DUB strategy, 2) the tighter AdaHedgeD strategy, and 3) the DORM+ algorithm that is tuning-free. This tuning-free property has evident practical benefits, as this section demonstrates.

Fig. 5.3 shows the yearly regret of the DUB, AdaHedgeD, and DORM+ algorithms. A consistent pattern appears in the yearly regret: DUB has moderate positive regret, AdaHedgeD has both the largest positive and negative regret values, and DORM+ sits between these two extremes. If we examine the weights played by each algorithm (Fig. 5.4), the weights of DUB and AdaHedgeD appear respectively over- and under-regularized compared to DORM+ (the top model for this task). DUB's use of the upper bound $\boldsymbol{b}_{t,F}$

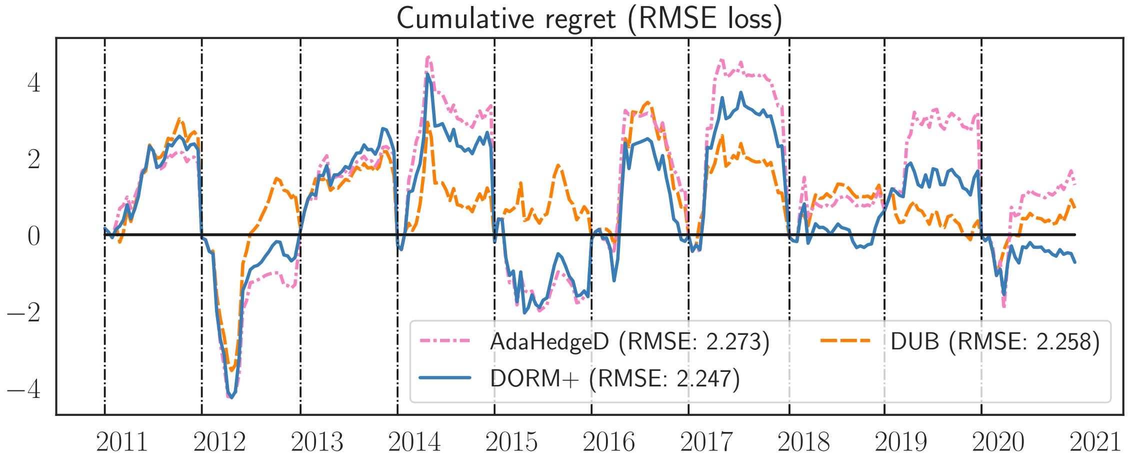


Figure 5.3: ***Regret of Regularizers****: Yearly cumulative regret (in terms of the RMSE loss) for the three regularization strategies for the Temp. 3-4w task.*

results in a very large regularization setting ($\lambda_T = 142.881$) and a virtually uniform weight setting. AdaHedgeD's tighter bound δ_t produces a value for $\lambda_T = 3.005$ that is two orders of magnitude smaller. However, in this short-horizon forecasting setting, AdaHedgeD's aggressive plays result in higher average RMSE. By nature of it's λ_t-free updates, DORM+ produces more moderately regularized plays $\boldsymbol{w}_t$ and negative regret.

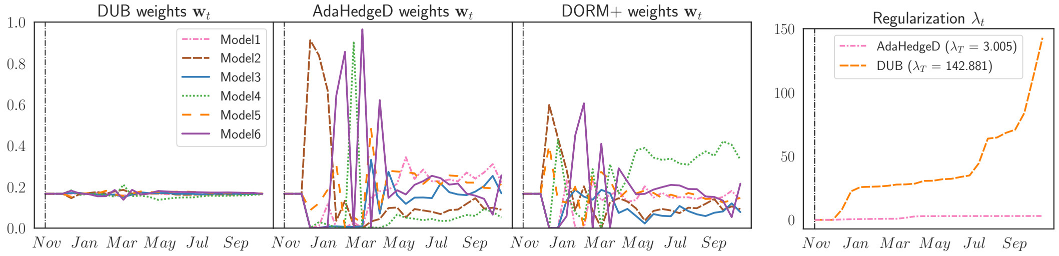


Figure 5.4: ***Impact of Regularization:*** *The plays $\mathbf{w}_t$ of online learning algorithms used to combine the input models for the Temp. 3-4w task in the 2020 evaluation year. The weights of* DUB *and* AdaHedgeD *appear respectively over and under regularized compared to* DORM+ *(the top model for this task) due to their selection of regularization strength λ_t (right).*

To replicate or not to replicate In this section, we compare the performance of replicated and non-replicated variants of our DORM+ algorithm. Both algorithms perform well (see App. C.12.3), but in all tasks, DORM+ outperforms replicated DORM+ (in which $D + 1$ independent copies of DORM+ make staggered predictions). Fig. 5.5 provides an example of the weight plots produced by the replication strategy in the Temp. 5-6w task with $D = 3$. The separate nature of the replicated learner's plays is evident in the weight plots and leads to an average RMSE of 2.315, versus 2.303 for DORM+ in the Temp. 5-6w task.

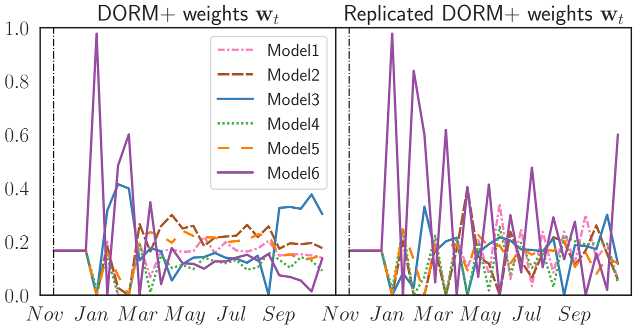


Figure 5.5: ***To Replicate or not to Replicate:*** *The plays $\boldsymbol{w}_t$ of standard* DORM+ *and replicated* DORM+ *algorithms for the Temp. 5-6w task in the final evaluation year.*

Learning to hint Finally, we examine the effect of optimism on the DORM+ algorithms and the ability of our "learning to hint" strategy to recover the performance of the best optimism strategy in retrospect. Following the hint construction protocol in App. C.13.2, we run the DORM+ base algorithm with $m = 4$ subgradient hinting strategies: $\tilde{\boldsymbol{g}}_s = \boldsymbol{g}_{t-D-1}$ (`recent_g`), $\tilde{\boldsymbol{g}}_s = \boldsymbol{g}_{s-D-1}$ (`prev_g`), $\tilde{\boldsymbol{g}}_s = \frac{D+1}{t-D-1}\boldsymbol{g}_{1:t-D-1}$ (`mean_g`), or $\tilde{\boldsymbol{g}}_s = \mathbf{0}$ (`none`). We also use DORM+ as the meta-algorithm for hint learning to produce the `learned` optimism strategy that plays a convex combination of the four hinters. In Fig. 5.6, we first note that several optimism strategies outperform the `none` hinter, confirming the value of optimism in reducing regret. The `learned` variant of DORM+ avoids the worst-case performance of the individual hinters in any given year (e.g., 2015), while staying competitive with the best strategy (although it does not outperform the dominant `recent_g` strategy overall). We believe the performance of the online hinter could be further improved by developing tighter convex bounds on the regret of the base problem in the spirit of Assump. 5.8.1.

5.10 Conclusion

In this work, we confronted the challenge of balancing exploration and exploitation in online learning with delayed feedback and short regret horizons. Making use of optimism, we developed practical non-replicated, self-tuned and tuning-free algorithms with optimal regret guarantees. Our "delay as optimism" reduction and our refined analysis of optimistic learning

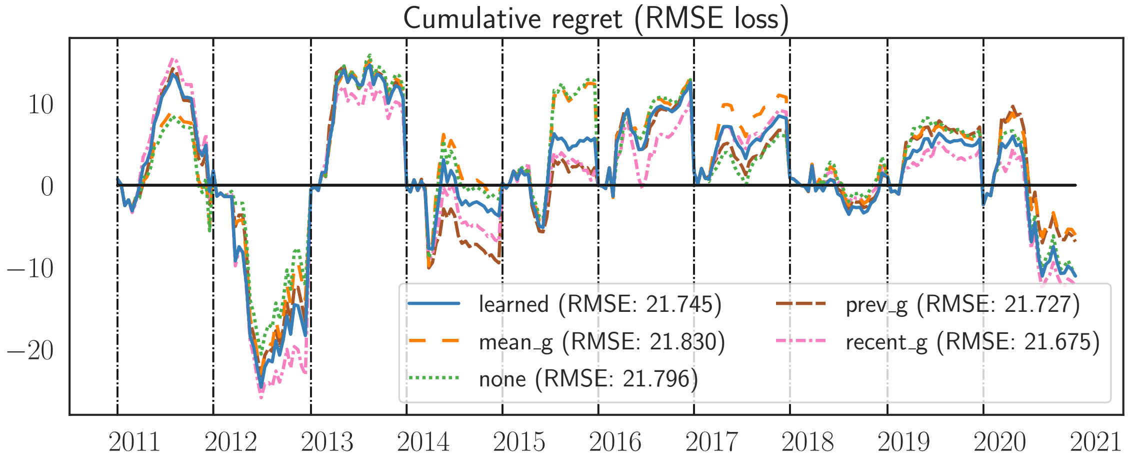


Figure 5.6: ***Learning to Hint****: Yearly cumulative regret (in terms of the RMSE loss) for the adaptive hinting and four constant hinting strategies for the Precip. 3-4w task.*

produced novel regret bounds for both optimistic and delayed online learning and elucidated the connections between these two problems. Within the subseasonal forecasting domain, we demonstrated that delayed online learning methods can produce zero-regret forecast ensembles that perform robustly from year-to-year. Our results highlighted DORM+ as a particularly promising candidate due to its tuning-free nature and small tracking regret.

Adversarial online learning paradigms, such as were explored in this chapter, have many potential applications for scientific problems, including model selection, adaptive ensembling, and parameter tuning. Unlike the POMDP formulations explored in the previous chapters, adversarial online learners make minimal assumptions about the nature of the loss sequence that the environment will produce. In an application such as weather forecasting, the underlying dynamics are fundamentally chaotic and putting a reasonable probability distribution over potential outcomes is challenging. State-of-the-art model-based dynamical forecasting models tend to underestimate real-world variability [96]. POMDP models, which maximize expected reward given a belief, assume the distribution of current and future state outcomes are captured accurately by the POMDP model. When these assumptions are violated, POMDP policies may exhibit very little robustness. Online learning methods, on the other hand, can provide regret guarantees under even adversarial realizations of the environment. Adapting online learning algorithms to the bandit setting, where only the loss of the selected action is observed, is often straightforward [94]. However, the bandit and online learning formulations have the disadvantage of being unable to reason explicitly about future outcomes, which limits their application to certain scientific domains, such as exploratory robotics, where predicting and planning with respect to future states and rewards is critical.

Chapter 6

Conclusions and Future Directions

Developing autonomous computational systems that can perform scientific discovery requires online decision-making under uncertainty. These decision-makers contend with an environment that is often modeled as being stochastic or adversarial. In the face of future outcomes that are difficult to predict and uncertain, balancing exploration and exploitation is a key issue for real-time decision-makers. This thesis considered the exploration and exploitation trade-off in two different sequential decision-making formulations, online learning and partially observable Markov decision processes, and demonstrated that carefully targeting a specific task or loss function in exploration can lead to an efficient balance between exploring and exploiting that enables planner performance guarantees and regret bounds.

6.1 Summary of Contributions

Chapters 3 and 4 dealt with the POMDP decision-making formulation and asked how a robot or autonomous agent could place sensors in the right place at the right time and collect the right data to inform model validation and development. In this formulation, exploration corresponded to taking sensing actions that reduce state uncertainty but may not make direct progress towards accomplishing an overall scientific task. Exploitation corresponded to using the current knowledge of the state to collect useful observations for the scientific objective. The key for balancing exploration and exploitation is the appropriate valuation of information: in the face of significant state uncertainty, an agent must recognize when reducing that uncertainty is essential for accomplishing a given task and when state uncertainty does not affect task performance. These chapters introduced two metrics for valuing information in decision-making contexts and used them to perform targeted exploration and sequential decision-making.

6.1.1 PLUMES: Source-Seeking with Targeted Information Rewards

Chapter 3 introduces the maximum-value information reward, a specific information measure about a random variable—the maximum value in an environment—that was essential for accomplishing a task. In these source-seeking applications, the task can be distilled to reducing the uncertainty in the distribution of a few key random variables, i.e., the location and value of the global maximum in the environment. We adapted the maximum-seeking POMDP to use this heuristic reward to guide a robotic Monte Carlo tree search planner and demonstrated that the maximum-value information reward drove the robot to collect observations that efficiently identified the global maximum. Compared to baseline algorithms that used, e.g,. UCB heuristics to balance exploration and exploitation, the PLUMES algorithm performed much less unnecessary exploration and was less likely to get stuck in a local maximum due to premature exploitation.

6.1.2 Value of Information and Macro-action Discovery in POMDPs

For the source-seeking problem, the valuation of information is somewhat straightforward. A human algorithm designer may identify that exploration is only necessary to reduce uncertainty about the key random variables of source location and value and design an information reward that trades of exploration and exploitation appropriately. This simple identification of key variables may not be possible in all tasks.

For example, consider the task of collecting samples buoyant stem of a deep sea hydrothermal plume, a variation of the problem considered in Chapter 3. The exact location of the buoyant stem in the water column changes in ways that can be difficult to predict precisely due to currents and turbulent mixing. What information is needed to accomplish this task? It is not as simple as requiring the state of the buoyant stem to be perfectly known. A specific robotic platform might actually move too slowly to track the buoyant stem location, even if that evolution could be predicted. It may be sufficient to constrain the evolution within a specific region and sample uniformly within that region. Or, it may be enough to identify the location of the plume during slack tide, when current effects are minimal. This complex interaction between information about state, the constraints of a specific decision-maker (e.g,. robotic platform), and the specification of the task inspired the approach in Chapter 4, which goes beyond the hand-designed information rewards of Chapter 3 to introduce a general value of information metric.

Chapter 4 introduces a value of information (VoI) metric that directly approximates the sensitivity of the POMDP value function to information gathering actions. By measuring the gap between the value (i.e., long term task performance) of taking an optimal, closed-loop action from a given belief state and taking an optimal, open-loop action (which does not provide an observation or reduce the uncertainty in the belief state), we can estimate how valuable information is from a given belief state. Because this method makes use of a value function approximation, it inherently accounts for factors such as decision-maker constraints and task specification. As an application of the utility of VoI in balancing exploration and exploitation in planning problems, we demonstrated that VoI can be used to construct open-loop macro-actions—sequences of actions taken in open-loop that can be used to reduce the effective horizon and complexity of planning. In a set of simulated tracking experiments, we demonstrated that these open-loop macroaction policies effectively balance exploration and exploitation in planning while maintaining performance guarantees with respect to the optimal closed-loop policy.

The two strategies introduced in Chapters 3 and 4—one task-specific and one general—demonstrated how specifically considering the question of the valuation of information can lead to efficient planning algorithms with performance guarantees.

6.1.3 Online Learning with Optimism and Delay

Chapter 5 considered exploration and exploitation in online learning problems, a different class of sequential decision-making problems in which a learner is pitted against an adversarial environment. Online learning forms the basis of model selection and adaptive ensembling in scientific applications. In this problem domain, the exploration and exploitation trade-off allows the learner to balance between trusting the predictability of the environment, given past experiences and forecasts for future states, while playing conservatively enough to maintain performance guarantees under adversarial outcomes.

Within the framework of online learning, we considered the dual questions of:

- In an adversarial environment, how should an online algorithm balance exploration and exploitation in the presence of delayed feedback?
- When the future state of the environment is partially predictable, how can an online learning leverage that predictability, while maintaining performance guarantees in the adversarial setting?

The key challenge in answering these questions is formulating algorithms that automatically tuned the balance between exploration and exploitation based on observed performance.

This chapter formulated three online learning algorithms—DORM, DORM+, and AdaHedgeD that automatically tuned their behavior to balance exploration and exploitation and demonstrated, via novel analysis methods, that the regret of these algorithms is rate-optimal. We demonstrated how these online learning algorithms can be used for real-world subseasonal temperature and precipitation forecasting.

6.2 Recommendations for Future Work

This thesis presented three algorithms for sequential decision-making and applied each to a real-world scientific problem. There are many promising areas of future work for both the algorithms and application areas presented. We highlight several in this section.

6.2.1 Learning the Value of Information

Chapter 4 develops an analytical but approximate calculation for value of information in planning problems. The experimental demonstration directly approximates the PWLC POMDP value function with a finite set of α-vectors and measures the gap in the value function under open- and closed-loop actions. As is true for all POMDP solvers, approximating the value function for large discrete or continuous state spaces is computationally infeasible. As this value function approximations degrades, the VoI estimate also degrades. In Chapter 4, we demonstrated VoI estimation for a problem with $|\mathcal{S}| = 10,000$, $|\mathcal{A}| = 5$, and $|\mathcal{Z}| = 100$. The limited size of this experimental demonstration was driven by the difficulty of efficient value function approximation for POMDPs.

Learning-based methods have shown promise for approximating the value function of fully observable or partially observable problems based on simulated agent experiences. Learning techniques could be used to approximate the POMDP value function and facilitate the computation of VoI by circumvent the computational burden of exact POMDP planning.

Alternatively, approximating whether a given belief state has high or low VoI may be an easier problem than learning the value of that belief state. A more promising approach may be to directly learn the value of information based on experience. Previous methods, such as Liu et al. [106], Stadler et al. [167], Stein et al. [168] have demonstrated that auxiliary metrics of planning success, such as likelihood of a given exploratory action succeeding, can be learned effectively from data. This style of learning framework may be a promising way of scaling VoI approximations to large and continuous problems.

6.2.2 Continuous POMDP Planning

Current state-of-the art POMDP planners have key limitations that make them poorly suited to address real-world scientific tasks. Offline POMDP solvers have largely been limited to small, discrete environments. Online POMDP solvers—which do allow for large continuous state—require an agent to simulate system dynamics and perform inference in real-time while planning. The computational cost of expensive dynamics models and inference procedures makes online planning intractable in many scientific problems. To alleviate the cost of online planners, we can use offline POMDP planners to approximate the POMDP value function and limited look-ahead online planners to fine-tune plans in real-time. However, this requires offline POMDP planners that can plan efficiently in continuous state, action, and observation spaces.

The POMDP objective—optimizing expected reward under uncertainty—gives rise to a value function that has been shown to have piecewise-linear and convex (PWLC) structure in discrete finite- and infinite-time planning problems [160]. In continuous domains, PWLC approximations of the value function are known only for certain classes of value function (e.g., mixture of Gaussians [140, 162]). Developing scalable value function approximations for large POMDPs that can leverage PWLC problem structure is a key step in improving both offline and online planning under uncertainty.

6.2.3 Predictability and Adaptive Horizon Planning

Chapter 4 exposed an interesting connection between the predictability of a phenomenon and the optimal planning strategy. When a phenomenon is highly predictable, “planning” is useful: an open- or closed-loop planner can forecast the state over long horizons and select high-reward actions. On the other hand, when a phenomenon is highly unpredictable, either due to highly stochastic dynamics or significant state uncertainty, planning may be less useful; the set of possible outcomes or contingencies which must be planned for is large. This complicated closed-loop planning, and requires open-loop planners that can perform well in average over all contingencies. Understanding the horizon over which a planner can be effective and computationally feasible, given the predictability of the environment and current state estimate, is a promising area for extending approaches such as the VoI approximation method presented in Chapter 4.

6.2.4 Efficient Information Planning in Sequential Problems

Applying information planning and optimal experimental design to real-world scientific problems requires combining information-theoretic objectives with sequential decision-making. Information rewards are known to be expensive to compute and often require approximate estimators [199]. Developing decision-makers that reason about information rewards may also require reasoning about the computational expense of obtaining those reward estimates and choosing when to perform that evaluation selectively, as in Zheng et al. [200]. Integrating information rewards into real-world, nonmyopic robotics planners with state requires extending these ideas to online or offline POMDP planners [44], and would enable scientific objectives to be specified in terms of information gained about a specific variable or feature of interest.

6.2.5 Real-World Scientific Deployments for Oceanography

Oceanographic research expeditions require chartering a research vessel that can house scientific teams for weeks at a time, facilitate deploying and recovering scientific instruments, and transit many thousands of kilometers for any given expedition. Depth-capable AUV assets, like AUV SENTRY operated by Woods Hole Oceanographic Institution (WHOI) and the National Deep Submergence Facility (NDSF) [78], may be deployed on tens of expeditions each year and operated approximately two-thirds of the year in total. A specialized team is typically deployed with the vehicle, responsible for vehicle maintenance and working with the scientific leader to design and execute missions while at sea.

Under these pressures, the efficiency and safety of AUV technologies like SENTRY are essential. These AUVs are typically flown in fixed survey patterns (e.g., lawnmowers) that can be easily verified before a dive and monitored while underway. Indeed, even rudimentary adaptive behaviors are not an inherent ability of many AUVs, although they may be possible with a human-in-the-loop. One of the difficulties in oceanographic science is the limited communication bandwidth that exists between ship and AUV. Underwater, acoustic modems are required. These are range-limited, can be lossy, and have severely limited bandwidth.

The algorithm in Chapter 3 assumed that closed-loop planning was possible. However, the communication constraints discussed mean that AUVs for deep sea research are often operated only in open-loop, and performing any closed-loop behaviors that are validated by the ship requires expensive acoustic data transfer. The extreme cost of communication means that algorithms such as those presented in Chapter 4 could be very valuable. Developing algorithms that enable an AUV to reason about when it requires an updated model and plan to accomplish a given task, and when the current model and plan are sufficient is a promising practical application of the ideas presented in this thesis. Any adaptation of these algorithmic ideas to real-world oceanographic deployment require careful consideration of the practical constraints of these vehicles and the ship environment.

6.3 Concluding Remarks

Observational data are our lens to the world around us. These data are a critical element in scientific discovery, providing the ability validate physical theories or informing the development of new models. The value of developing computational systems that can enhance human-driven scientific experimentation and data collection is evident; this value has already been realized in several application areas, from animal behavior monitoring to renewable material synthesis to oceanographic exploration [45, 51, 55, 58, 198]. The thesis identifies the exploration-exploitation trade-off as one of the key bottlenecks in realizing autonomous scientific decision-making and proposes task-targeted exploration—directly quantifying sensitivity of task performance to exploratory actions—as a potential solution. The thesis then leverages task-targeted exploration to present three methodological contributions in decision-making under uncertainty that are motivated by and applied to real-world scientific applications. Each demonstrates how autonomous decision-making can enhance human scientific discovery in complex domains by reasoning over noisy and heterogeneous sensors and chaotic or stochastic structure to place a sensor in the right place at the right time to validate a scientific hypothesis or to produce an accurate forecast [1]

[1] This material is based upon work supported by the NSF Graduate Research Fellowship Program and a Microsoft Research PhD Fellowship, as well as the Department of Energy / National Nuclear Security Administration under Award Number DE-NA0003921, the Office of Naval Research under Award Number N00014-17-1-2072, and DARPA under Award Number HR001120C0033.

Appendix A

PLUMES: Source-Seeking with Targeted Information Rewards

This appendix includes additional analysis not presented in the main text for clarity of the exposition. App. A.1 introduces standard information measures and App. A.2 includes a description of the rank-1 matrix updates used for streaming Gaussian processes in PLUMES.

A.1 Information Measures

In fields such as experimental design and informative path planning, the objective of the agent is often to optimize some measure of "information" collected about a variable of interest. There are several common measures from the field of information theory that can be used to quantity the uncertainty or uncertainty reduction in a random variable. These fundamental quantities are covered in detail in, e.g., Cover and Thomas [35]. Chapter 3 makes use of mutual information for continuous random variables to choose promising sample points in an underlying Gaussian process model.

Entropy and Differential Entropy Entropy is commonly used to measure the uncertainty of a random variable. For discrete random variable $X \in \mathcal{X}$, the Shannon entropy is defined as:

$$H(X) \triangleq -\sum_{x \in \mathcal{X}} \mathbb{P}(X = x) \log \mathbb{P}(X = x) \tag{A.1}$$

Given a second random variable $Z \in \mathcal{Z}$ that has joint distribution $\mathbb{P}(X, Z)$ with X, we can define the joint entropy as:

$$H(X, Z) \triangleq -\sum_{z \in \mathcal{Z}} \sum_{z in \mathcal{X}} \mathbb{P}(X = x, Z = z) \log \mathbb{P}(X = x, Z = z) \tag{A.2}$$

We can condition the variable X on a Z taking value $z \in \mathcal{Z}$ and define the entropy:

$$H(X \mid Z = z) \triangleq -\sum_{x \in \mathcal{X}} \mathbb{P}(X = x \mid Z = z) \log \mathbb{P}(X = x \mid Z = z). \tag{A.3}$$

Finally, averaging over different values of Z gives the conditional Shannon entropy:

$$\begin{aligned} H(X \mid Z) &\triangleq \sum_{z \in \mathcal{Z}} \mathbb{P}(Z = z) H(X \mid Z = z) && \text{(A.4)} \\ &= -\sum_{z \in \mathcal{Z}} P(Z = z) \sum_{x \in \mathcal{X}} \mathbb{P}(X = x \mid Z = z) \log \mathbb{P}(X = x \mid Z = z). && \text{(A.5)} \end{aligned}$$

This expression is connected to the joint entropy by the following expression $H(X, Z) = H(X \mid Z) + H(Z) = H(Z \mid X) + H(X)$.

For continuous random variable $Y \in \mathcal{Y}$ with probability density function $f(y)$, we can define a continuous analog of Shannon entropy known as *differential entropy*:

$$H(Y) \triangleq -\int_{y \in \mathcal{Y}} f(y) \log f(y) \, \mathrm{d}y. \tag{A.6}$$

Although this is a straightforward extension of Shannon entropy, it does not share many of the appealing properties of the discrete expression. For example, differential entropy may be negative and is not invariant under invertible mappings such as rescaling.

Mutual Information For two random variables $X \in \mathcal{X}$ and $Z \in \mathcal{Z}$ related by joint distribution $\mathbb{P}(X, Z)$, mutual information measures the entropy reduction in one random variable caused by observation of the other. It is often defined as a difference in entropies:

$$I(X; Z) \triangleq H(X) - H(X \mid Z) = H(Z) - H(Z \mid X). \tag{A.7}$$

This can be expressed directly in terms of the joint probability distribution:

$$I(X; Z) = \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{Z}} \mathbb{P}(X = x, Z = z) \log \frac{\mathbb{P}(X = x, Y = y)}{\mathbb{P}(X = x)\mathbb{P}(Y = y)}. \tag{A.8}$$

For continuous random variables, the mutual information expression is exactly analogous and is better behaved than differential entropy: for both discrete and continuous random variables, mutual information is positive and invariant under invertible transformations of the input random variables.

A.2 Rank One Streaming Updates for Gaussian Processes

Generally, a Gaussian process (GP) model is parameterized by a prior mean $\mu_p(\boldsymbol{x})$ and covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$. In the following, we assume the prior mean function is the zero function. In the basic GP model, the measurement function is also Gaussian with some fixed measurement noise σ_n^2.

Given a dataset of observation location and observation pairs, $\mathcal{D} = \{\boldsymbol{x}_i, z_i\}_{i=0}^{D-1}$ of size D, the posterior predictive of the function f at a new location $\boldsymbol{x}' \in \mathbb{X}$ is given by the following closed-form solution:

$$f(\boldsymbol{x}') \mid \mathcal{D} \sim \mathcal{N}(\mu(\boldsymbol{x}'), \sigma^2(\boldsymbol{x}')), \text{where} \tag{A.9}$$

$$\mu(\boldsymbol{x}') = \kappa(\boldsymbol{x}')^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \boldsymbol{z}, \tag{A.10}$$

$$\sigma^2(\boldsymbol{x}') = \kappa(\boldsymbol{x}', \boldsymbol{x}') - \kappa(\boldsymbol{x}')^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \kappa(\boldsymbol{x}'), \tag{A.11}$$

where $\boldsymbol{z} = [z_0, \ldots, z_{D-1}]^\top$, $\mathbf{K} \in \mathbb{S}_+$ is the positive semi-definite kernel matrix with $\mathbf{K}[i,j] = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}$, and $\kappa(\boldsymbol{x}') = [\kappa(\boldsymbol{x}_0, \boldsymbol{x}'), \ldots, \kappa(\boldsymbol{x}_{D-1}, \boldsymbol{x}')]^\top$.

This simple, closed-form solution for the posterior predictive distribution enables the mean and variance of the value of the environment at a new location to be predicted efficiently, given a moderately-sized, fixed dataset $\mathcal{D}$. However, evaluating (A.10) and (A.11) require inverting the kernel matrix, which can become computationally expensive (naively requires $\mathcal{O}(D^3)$ operations; optimized libraries can provide $\approx \mathcal{O}(D^{2.8074})$ improved complexity [170]).

In this section, we focus on the challenges of incorporating streaming data into a Gaussian process model. Streaming data arise commonly in robotics and online applications. Each time a new data point is added to the set $\mathcal{D}$, a corresponding row and column are added to the $D \times D$ kernel matrix. A naive implementation of a streaming Gaussian process model would re-invert the kernel matrix at each timestep, despite the matrix only changing by a single data point. Several techniques exist for scaling GP models to large, streaming datasets [103] by approximating the kernel matrix. Here, we describe a simple, exact approach based on the matrix inversion lemma.

Lemma A.2.1 (Matrix Inversion Lemma, Rasmussen and Williams [148], Section A.3). *Let $\mathbf{Z}$ be an $n \times n$ matrix, $\mathbf{W}$ be $m \times m$, and $\mathbf{U}$ and $\mathbf{V}$ be of size $n \times m$. Assume $\mathbf{Z}^{-1}$ is known and then $\mathbf{Z}$ undergoes a low-rank perturbation, i.e., $\mathbf{Z}' = \mathbf{Z} + \mathbf{U}\mathbf{W}\mathbf{V}^\top$, where $m < n$. Then the Woodbury, Sherman, & Morrison formula allows the updated inverse ${\mathbf{Z}'}^{-1}$ to be* *computed efficiently:*

$$(\mathbf{Z} + \mathbf{U}\mathbf{W}\mathbf{V}^\top)^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{U}(\mathbf{W}^{-1} + \mathbf{V}^\top\mathbf{Z}^{-1}\mathbf{U})^{-1}\mathbf{V}^\top\mathbf{Z}^{-1}. \tag{A.12}$$

This formula can be extended to invert a partitioned matrix.

Lemma A.2.2 (Inverse of a Partitioned Matrix (see, e.g., Rasmussen and Williams [148], Section A.3)**.** *Let the $n \times n$ matrix $\mathbf{A}$ be invertible. Consider the block partitions:*

$$\mathbf{A} = \begin{pmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{pmatrix}, \qquad \mathbf{A}^{-1} = \begin{pmatrix} \tilde{\mathbf{P}} & \tilde{\mathbf{Q}} \\ \tilde{\mathbf{R}} & \tilde{\mathbf{S}}, \end{pmatrix} \tag{A.13}$$

where $\mathbf{P}$ and $\tilde{\mathbf{P}}$ are $n_1 \times n_1$ matrices and $\mathbf{S}$ and $\tilde{\mathbf{S}}$ are $n_2 \times n_2$ matrices, $n = n_1 + n_2$.

Define $\mathbf{M} = (\mathbf{S} - \mathbf{R}\mathbf{P}^{-1}\mathbf{Q})^{-1}$. Then, the block entries of the matrix $\mathbf{A}^{-1}$ are given by:

$$\tilde{\mathbf{P}} = \mathbf{P}^{-1} + \mathbf{P}^{-1}\mathbf{Q}\mathbf{M}\mathbf{R}\mathbf{P}^{-1} \tag{A.14}$$

$$\tilde{\mathbf{Q}} = -\mathbf{P}^{-1}\mathbf{Q}\mathbf{M} \tag{A.15}$$

$$\tilde{\mathbf{R}} = -\mathbf{M}\mathbf{R}\mathbf{P}^{-1} \tag{A.16}$$

$$\tilde{\mathbf{S}} = \mathbf{M} \tag{A.17}$$

An equivalent formulation is given by defining $\mathbf{N} = (\mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R})^{-1}$, resulting in:

$$\tilde{\mathbf{P}} = \mathbf{N} \tag{A.18}$$

$$\tilde{\mathbf{Q}} = -\mathbf{N}\mathbf{Q}\mathbf{S}^{-1} \tag{A.19}$$

$$\tilde{\mathbf{R}} = -\mathbf{S}^{-1}\mathbf{R}\mathbf{N} \tag{A.20}$$

$$\tilde{\mathbf{S}} = \mathbf{S}^{-1} + \mathbf{S}^{-1}\mathbf{R}\mathbf{N}\mathbf{Q}\mathbf{S}^{-1} \tag{A.21}$$

$$\tag{A.22}$$

Consider receiving one additional, streaming observation at location $\boldsymbol{x'}$ in a Gaussian process model with kernel matrix and inverse given by $\mathbf{K}$ and $\mathbf{K}^{-1}$ respectively. The new kernel matrix is formed by adding an additional row and column to $\mathbf{K}$:

$$\mathbf{K}' = \begin{pmatrix} \mathbf{K} & \kappa(\boldsymbol{x'}) \\ \kappa(\boldsymbol{x'})^\top & \kappa(\boldsymbol{x'}, \boldsymbol{x'}), \end{pmatrix} \tag{A.23}$$

where $\kappa(\boldsymbol{x'})$ is a rank-one matrix and $\kappa(\boldsymbol{x'}, \boldsymbol{x'})$ is a scalar, as defined in (A.10) and (A.11).

The updated kernel inverse $\mathbf{K'}^{-1}$ can be computed using the first formulation in Lem. A.2.2, with:

$$\mathbf{P} = \mathbf{K} \tag{A.24}$$

$$\mathbf{Q} = \kappa(\boldsymbol{x'}) \tag{A.25}$$

$$\mathbf{R} = \kappa(\boldsymbol{x'})^\top \tag{A.26}$$

$$\mathbf{S} = \kappa(\boldsymbol{x'}, \boldsymbol{x'}), \tag{A.27}$$

resulting in a value of $\mathbf{M}$ that is a scalar value.

Appendix B

Value of Information and Macro-action Discovery in POMDPs

This appendix includes additional analysis not presented in the main text for clarity of the exposition. Section B.1 includes detailed derivations of Lemma 5.3, 5.4 and 5.5.

B.1 Analysis

The following section contains the detailed derivation of Lemmas 5.3-5.5 presented in the main text.

B.1.1 Value Backup Error

Lemma 5.3 *The horizon-H value function error caused by including open-loop actions in backups whenever VoI $< \tau$ is bounded for beliefs in $\mathcal{G}$ by $\epsilon_H = \left\| \hat{V}_H^* - V_H^* \right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}(2L\delta_{\mathcal{B}} + \tau)$.*

Proof. Consider any compact subset β of $\mathcal{G}$; importantly, β can be different from the belief set $\mathcal{B}$ used in planning. We define ϵ_h to be the maximum error in the value function on the set β during the value iteration recursion at horizon h:

$$\epsilon_h = \left\| V_h^*(\beta) - \hat{V}_h^*(\beta) \right\|_\infty, \tag{B.1}$$

where we use the notation $f(\beta)$ to denote the restriction of the function f to the domain β. Then:

$$\epsilon_h = \left\| \mathcal{H}V_{h-1}^*(\beta) - \hat{\mathcal{H}}\hat{V}_{h-1}^*(\beta) \right\|_\infty, \tag{B.2}$$

Let $b_\epsilon \in \beta$ be the belief for which the value function error is maximized:

$$b_\epsilon = \arg\max_{b\in\beta} \left\| \mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\|. \tag{B.3}$$

Let δ be the minimum distance between a belief in $\mathcal{B}$ and b_ϵ: $\delta = \min_{b\in\mathcal{B}}\|b - b_\epsilon\|_1$ and let belief $b \in \mathcal{B}$ be a minimizer. Because $\mathcal{B}$ forms a $\delta_\mathcal{B}$ covering of β, we have that $\delta \leq \delta_\mathcal{B}$.

We bound ϵ_h as follows:

$$\begin{aligned}
\epsilon_h &= \left\| V^*_h(\beta) - \hat{V}^*_h(\beta) \right\|_\infty && \text{(B.4)}\\
&= \left\| \mathcal{H}V^*_{h-1}(b_\epsilon) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b_\epsilon) \right\| && \text{(B.5)}\\
&= \left\| \mathcal{H}V^*_{h-1}(b_\epsilon) - \mathcal{H}V^*_{h-1}(b) + \mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b_\epsilon) + \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\| && \text{(B.6)}\\
&\leq \left\| \mathcal{H}V^*_{h-1}(b_\epsilon) - \mathcal{H}V^*_{h-1}(b) \right\| + && \text{(B.7)}\\
&\qquad \left\| \hat{\mathcal{H}}\hat{V}^*_{h-1}(b_\epsilon) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\| + \left\| \mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\| \\
&\leq 2L\|b_\epsilon - b\|_1 + \left\| \mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\|. && \text{(B.8)}\\
&\leq 2L\delta_\mathcal{B} + \left\| \mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b) \right\|. && \text{(B.9)}
\end{aligned}$$

The term $2L\delta_\mathcal{B}$ represents the value-function error induced by the point-based approximation. We will further examine the term $\mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b)$, which represents the value function error due to the inclusion of potentially suboptimal macro-actions in the approximate backup operator.

Without loss of generality, let a_1 be the optimal action at belief b and a_2 be the near-optimal, open-loop action selected for backing up $\hat{V}^*_h$. Let $\mathcal{H}^{a_1}$ denote the standard value function backup using action a_1 and $\hat{\mathcal{H}}^{a_{2,OLP}}$ denote the macro-action backup using action a_2 in open loop.

$$\begin{aligned}
\left\|\mathcal{H}V^*_{h-1}(b) - \hat{\mathcal{H}}\hat{V}^*_{h-1}(b)\right\| &= \left\|\mathcal{H}^{a_1}V^*_{h-1}(b) - \hat{\mathcal{H}}^{a_2,OLP}\hat{V}^*_{h-1}(b)\right\| && \text{(B.10)}\\
&\leq \left\|\mathcal{H}^{a_2}V^*_{h-1}(b) + \tau - \hat{\mathcal{H}}^{a_2,OLP}\hat{V}^*_{h-1}(b)\right\| && \text{(B.11)}\\
&\leq \left\|\mathcal{H}^{a_2,OLP}V^*_{h-1}(b) + \tau - \hat{\mathcal{H}}^{a_2,OLP}\hat{V}^*_{h-1}(b)\right\| && \text{(B.12)}\\
&\leq \|\mathbb{E}_{s\sim b}[R(s,a_2)] + \gamma V^*_{h-1}(b^{a_2,*}) + \tau - && \text{(B.13)}\\
&\qquad \mathbb{E}_{s\sim b}[R(s,a_2)] - \gamma\hat{V}^*_{h-1}(b^{a_2,*})\| \\
&\leq \left\|\gamma V^*_{h-1}(b^{a_2,*}) + \tau - \gamma\hat{V}^*_{h-1}(b^{a_2,*})\right\| && \text{(B.14)}\\
&\leq \gamma\epsilon_{h-1} + \tau, && \text{(B.15)}
\end{aligned}$$

where if $b^{a_2,*} \notin \mathcal{G}$, we replace $V^*_{h-1}(b^{a_2,*})$, $\hat{V}^*_{h-1}(b^{a_2,*})$ with a valid lower-bound.

Because $V^*_0 \equiv \hat{V}^*_0$, we have that $\epsilon_0 = 0$. Expanding the recursion $\epsilon_h \leq \gamma\epsilon_{h-1} + 2L\delta_{\mathcal{B}} + \tau$, we conclude that $\epsilon_H \leq \frac{1-\gamma^H}{1-\gamma}(2L\delta_{\mathcal{B}} + \tau)$. ■

B.1.2 Interpolating Macro-Actions

Lemma 5.4 *(Lasota and Mackey [93]) The open-loop dynamics are a non-expansive mapping in belief space. Consider two beliefs $b_1, b_2 \in \Pi(\mathcal{S})$ such that $\|b_1 - b_2\|_1 = \delta$. Then, for any action a taken in open-loop, it follows that $\left\|b_1^{a,*} - b_2^{a,*}\right\|_1 \leq k\delta$ for $0 \leq k \leq 1$.*

Proof. Following [93], let T_a be the Markov operator corresponding to the POMDP open-loop transition dynamics under action a:

$$\begin{aligned}
\left\|b_1^{a,*} - b_2^{a,*}\right\|_1 &= \|T_a b_1 - T_a b_2\|_1 && \text{(B.16)}\\
&= \|T_a(b_1 - b_2)\|_1 && \text{(B.17)}\\
&\leq k\|b_1 - b_2\|_1 = k\delta, && \text{(B.18)}
\end{aligned}$$

where $0 \leq k \leq 1$ is the maximum contraction coefficient over actions $a \in \mathcal{A}$ and Eq. B.18 is a property of Markov operators [93]. ■

Lemma 5.5 *The additional value function error of approximating the VoI macro-action at belief b using its nearest neighbor b_* under k-contractive open-loop dynamics is bounded by:*

$$\eta_H = \left\|\hat{V}^*_H - V^{MA}_H\right\|_\infty \leq \frac{1-\gamma^H}{1-\gamma}\Big(L\delta_{\mathcal{B}} + \frac{R_{max}\delta_{\mathcal{B}}}{1-\gamma k} + L\gamma k\delta_{\mathcal{B}}\Big). \qquad \text{(B.19)}$$

Proof. Consider any compact subset β of $\mathcal{G}$. Define η_h to be the maximum difference between the value when utilizing optimal open-loop actions when VoI is low $\hat{V}_h^*$ and the value when performing macro-action chaining and interpolation V_h^{MA}:

$$\eta_h = \left\| \hat{V}_h^*(\beta) - V_h^{MA}(\beta) \right\|_\infty \tag{B.20}$$

Without loss of generality, let b be the belief for which Eq. B.20 is maximized, let b_* be it's nearest neighbor in $\mathcal{B}$, and let $A_l = \{a_1, \ldots, a_l\}$ be the length-l macro-action that is optimal at b_*.

$$\begin{aligned}
\eta_h &= \left\| \hat{V}_h^*(b) - V_h^{MA}(b) \right\|, && \text{(B.21)}\\
&\le \left\| \hat{V}_h^*(b) - \hat{V}_h^*(b_*) \right\| + \left\| \hat{V}_h^*(b_*) - V_h^{MA}(b) \right\|, && \text{(B.22)}\\
&\le L\delta_{\mathcal{B}} + \Bigg\| \sum_{i=0}^{l-1} \gamma^i \mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)] + \gamma^l \hat{V}_{h-l}^*(b_*^{A_{1:l}}) && \text{(B.23)}\\
&\qquad - \sum_{i=0}^{l-1} \gamma^i \mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)] - \gamma^l V_{h-l}^{MA}(b^{A_{1:l}}) \Bigg\|, \\
&\le L\delta_{\mathcal{B}} + \left\| \sum_{i=0}^{l-1} \gamma^i \Big(\mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)] - \mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)] \Big) \right\| && \text{(B.24)}\\
&\qquad\qquad + \gamma^l \left\| \hat{V}_{h-l}^*(b_*^{A_{1:l}}) - V_{h-l}^{MA}(b^{A_{1:l}}) \right\|. \\
&\le L\delta_{\mathcal{B}} + \left\| \sum_{i=0}^{l-1} \gamma^i \Big(\mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)] - \mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)] \Big) \right\| && \text{(B.25)}\\
&\qquad + \gamma^l \Big(\left\| \hat{V}_{h-l}^*(b_*^{A_{1:l}}) - \hat{V}_{h-l}^*(b^{A_{1:l}}) \right\| + \left\| \hat{V}_{h-l}^*(b^{A_{1:l}}) - V_{h-l}^{MA}(b^{A_{1:l}}) \right\| \Big) \\
&\le L\delta_{\mathcal{B}} + \left\| \sum_{i=0}^{l-1} \gamma^i \Big(\mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)] - \mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)] \Big) \right\| + \gamma^l (Lk^l \delta_{\mathcal{B}} + \eta_{h-l}) && \text{(B.26)}
\end{aligned}$$

where Eqs. B.22 and B.25 follow by the triangle inequality, Eq. B.23 using the fact that A_l is the optimal macro-action for b_* and will be applied to b under the macro-action policy, and Eq. B.26 by the contractive property of the open loop dynamics.

The form of Eq. B.23 reflects the expected reward when following the macro-action A_l

from both belief b and b_* and then reverting to the macro-action policy for the remainder of the horizon from the resulting belief. We can bound the final term in Eq. B.26 by further application of the non-expansive property to rewards collected during macro-action execution:

$$\left\|\sum_{i=0}^{l-1}\gamma^i\Big(\mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)]-\mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)]\Big)\right\| \tag{B.27}$$

$$\leq \sum_{i=0}^{l-1}\gamma^i\left\|\mathbb{E}_{s\sim b_*^{A_{1:i}}}[R(s,a_i)]-\mathbb{E}_{s\sim b^{A_{1:i}}}[R(s,a_i)]\right\| \tag{B.28}$$

$$\leq \sum_{i=0}^{l-1}\gamma^i\int_{\mathcal{S}}\left\|R(s,a_i)(b_*^{A_{1:i}}(s)-b^{A_{1:i}}(s))\right\|\mathrm{d}s \tag{B.29}$$

$$\leq \sum_{i=0}^{l-1}\gamma^i R_{max}\int_{\mathcal{S}}\left\|(b_*^{A_{1:i}}(s)-b^{A_{1:i}}(s))\right\|\mathrm{d}s \tag{B.30}$$

$$\leq \sum_{i=0}^{l-1}\gamma^i R_{max}\left\|b_*^{A_{1:i}}-b^{A_{1:i}}\right\|_1 \tag{B.31}$$

$$\leq \sum_{i=0}^{l-1}\gamma^i R_{max}k^i\delta=\frac{1-\gamma^l k^l}{1-\gamma k}R_{max}\delta \tag{B.32}$$

Plugging this expression in to Eq. B.23, we have the recursion: $\eta_h \leq L\delta_{\mathcal{B}}+\frac{1-\gamma^l k^l}{1-\gamma k}R_{max}\delta_{\mathcal{B}}+\gamma^l L k^l\delta_{\mathcal{B}}+\gamma^l\eta_{h-l}$. This expression depends on l, the length of the optimal macro-action at horizon h, in a complex way. Because l is variable and unknown *a priori*, we replace l with its case value in each expression: $\eta_h \leq L\delta_{\mathcal{B}}+\frac{R_{max}\delta_{\mathcal{B}}}{1-\gamma k}+\gamma Lk\delta_{\mathcal{B}}+\gamma\eta_{h-1}$. The result follows by expanding this recursion with $\eta_0=0$. ■

Appendix C

Online Learning with Optimism and Delay

C.1 Extended Literature Review

We review here additional prior work not detailed in the main thesis text.

C.1.1 General online learning

We recommend the monographs of Orabona [129], Shalev-Shwartz [153] and the textbook of Cesa-Bianchi and Lugosi [26] for surveys of the field of online learning and Joulani et al. [76], McMahan [115] for widely applicable and modular analyses of online learning algorithms.

C.1.2 Online learning with optimism but without delay

Syrgkanis et al. [176] analyzed optimistic FTRL and two-step variant of optimistic MD without delay. The work focuses on a particular form of optimism (using the last observed subgradient as a hint) and shows improved rates of convergence to correlated equilibria in multiplayer games. In the absence of delay, Steinhardt and Liang [169] combined optimism and adaptivity to obtain improvements over standard optimistic regret bounds.

C.1.3 Online learning with delay but without optimism

Overview Joulani et al. [74, 75], McMahan and Streeter [114] provide broad reviews of progress on delayed online learning.

Delayed stochastic optimization Agarwal and Duchi [1], Liu and Wright [104], Liu et al. [105], Nesterov [126], Recht et al. [149], Sra et al. [164] studied the effects of delay on stochastic optimization but do not treat the adversarial setting studied here.

FTRL-Prox vs. FTRL Joulani et al. [75] analyzed the delayed feedback regret of the *FTRL-Prox* algorithm, which regularizes toward the last played iterate as in online mirror descent, but did not study the standard FTRL algorithms (sometimes called *FTRL-Centered*) analyzed in this work.

C.1.4 Self-tuned online learning without delay or optimism

In the absence of optimism and delay, de Rooij et al. [39], Koolen et al. [85], Orabona and Pál [130] developed alternative variants of FTRL algorithms that self-tune their learning rates.

C.1.5 Online learning without delay for climate forecasting

Monteleoni et al. [121] applied the Learn-α online learning algorithm of Monteleoni and Jaakkola [120] to the task of ensembling climate models. The authors considered historical temperature data from 20 climate models and tracked the changing sequence of which model predicts best at any given time. In this context, the algorithm used was based on a set of generalized Hidden Markov Models, in which the identity of the current best model is the hidden variable and the updates are derived as Bayesian updates. This work was extended to take into account the influence of regional neighboring locations when performing updates [116]. These initial results demonstrated the promise of applying online learning to climate model ensembling, but both methods rely on receiving feedback without delay.

C.2 Proof of Thm. 5.4.1: OFTRL regret

We will prove the following more general result for optimistic adaptive FTRL (OAFTRL)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_{1:t} + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{w} \rangle + \lambda_{t+1} \psi(\boldsymbol{w}), \tag{OAFTRL}$$

from which Thm. 5.4.1 will follow with the choice $\lambda_t = \lambda$ for all $t \geq 1$.

Theorem C.2.1 (OAFTRL regret)**.** *If ψ is nonnegative and $(\lambda_t)_{t \geq 1}$ is non-decreasing, then,*

$\forall \boldsymbol{u} \in \mathbb{W}$, *the* OAFTRL *iterates* $\boldsymbol{w}_t$ *satisfy,*

$$\text{Regret}_T(\boldsymbol{u}) \le \lambda_T \psi(\boldsymbol{u}) + \sum_{t=1}^{T} \delta_t \tag{C.1}$$

$$\le \lambda_T \psi(\boldsymbol{u}) + \tag{C.2}$$

$$\sum_{t=1}^{T} \min\big(\frac{1}{\lambda_t}\text{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*), \text{diam}(\mathbb{W}) \min(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*)\big) \tag{C.3}$$

for

$$\delta_t \triangleq \min(F_{t+1}(\boldsymbol{w}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t), \quad \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \bar{\boldsymbol{w}}_t\rangle, \tag{C.4}$$

$$F_{t+1}(\hat{\boldsymbol{w}}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t\rangle)_+ \quad \textit{with} \tag{C.5}$$

$$\bar{\boldsymbol{w}}_t \triangleq \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t), \quad F_{t+1}(\boldsymbol{w}, \lambda_t) \triangleq \lambda_t \psi(\boldsymbol{w}) + \langle \boldsymbol{g}_{1:t}, \boldsymbol{w}\rangle, \quad \textit{and} \tag{C.6}$$

$$\hat{\boldsymbol{w}}_t \triangleq \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \lambda_t \psi(\boldsymbol{w}) + \langle \boldsymbol{g}_{1:t} + \min(\frac{\|\boldsymbol{g}_t\|_*}{\|\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t\|_*}, 1)(\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t), \boldsymbol{w}\rangle. \tag{C.7}$$

Proof. Consider a sequence of arbitrary auxiliary subgradient hints $\tilde{\boldsymbol{g}}_1^*, \dots, \tilde{\boldsymbol{g}}_T^* \in \mathbb{R}^d$ and the auxiliary OAFTRL sequence

$$\boldsymbol{w}_{t+1}^* = \operatorname*{arg\,min}_{\boldsymbol{w}^* \in \mathbb{W}} \langle \boldsymbol{g}_{1:t} + \tilde{\boldsymbol{g}}_{t+1}^*, \boldsymbol{w}^*\rangle + \lambda_{t+1}\psi(\boldsymbol{w}^*) \tag{C.8}$$

$$\text{for} \quad 0 \le t \le T \quad \text{with} \quad \tilde{\boldsymbol{g}}_{T+1}^* \triangleq \boldsymbol{0} \quad \text{and} \quad \lambda_{T+1} = \lambda_T. \tag{C.9}$$

Generalizing the forward regret decomposition of [76] and the prediction drift decomposition of [75], we will decompose the regret of our original $(\boldsymbol{w}_t)_{t=1}^T$ sequence into the regret of the auxiliary sequence $(\boldsymbol{w}_t^*)_{t=1}^T$ and the drift between $(\boldsymbol{w}_t)_{t=1}^T$ and $(\boldsymbol{w}_t^*)_{t=1}^T$.

For each time t, define the auxiliary optimistic objective function $\tilde{F}_t^*(\boldsymbol{w}) = F_t(\boldsymbol{w}) +$

$\langle \tilde{\boldsymbol{g}}_t^*, \boldsymbol{w} \rangle$. Fixing any $\boldsymbol{u} \in \mathbb{W}$, we have the regret bound

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) &\leq \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{u} \rangle && \text{(C.10)} \\
&\text{(since each } \ell_t \text{ is convex with } \boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{w}_t)) && \text{(C.11)} \\
&= \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}_t^* \rangle}_{\text{drift}} + \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t^* - \boldsymbol{u} \rangle}_{\text{auxiliary regret}}. && \text{(C.12)}
\end{aligned}$$

To control the drift term we employ the following lemma, proved in App. C.2.1, which bounds the difference between two OAFTRL optimizers with different losses but common regularizers.

Lemma C.2.1 (OAFTRL difference bound)**.** *The* OAFTRL *and auxiliary* OAFTRL *iterates* (C.8)*,* $\boldsymbol{w}_t$ *and* $\boldsymbol{w}_t^*$*, satisfy*

$$\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\| \leq \min\left(\frac{1}{\lambda_t} \|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_*, \text{diam}(\mathbb{W})\right). \tag{C.13}$$

Letting $a = \text{diam}(\mathbb{W}) \in \mathbb{R} \cup \{\infty\}$, we now bound each drift term summand using the Fenchel-Young inequality for dual norms and Lem. C.2.1:

$$\langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}_t^* \rangle \leq \|\boldsymbol{g}_t\|_* \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\| \leq \min\left(\frac{1}{\lambda_t} \|\boldsymbol{g}_t\|_* \|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_*, a\|\boldsymbol{g}_t\|_*\right). \tag{C.14}$$

To control the auxiliary regret, we begin by invoking the OAFTRL regret bound of [129, proof of Thm. 7.28], the nonnegativity of ψ, and the assumption that $(\lambda_t)_{t \geq 1}$ is non-decreasing:

$$\begin{aligned}
&\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t^* - \boldsymbol{u} \rangle && \text{(C.15)} \\
&\quad \leq \lambda_{T+1} \psi(\boldsymbol{u}) - \lambda_1 \psi(\boldsymbol{w}_1^*) + \sum_{t=1}^{T} F_{t+1}(\boldsymbol{w}_t^*, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) + (\lambda_t - \lambda_{t+1}) \psi(\boldsymbol{w}_{t+1}^*) && \text{(C.16)} \\
&\quad \leq \lambda_{T+1} \psi(\boldsymbol{u}) - \lambda_1 \psi(\boldsymbol{w}_1^*) + \sum_{t=1}^{T} F_{t+1}(\boldsymbol{w}_t^*, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t). && \text{(C.17)}
\end{aligned}$$

We next bound the summands in this expression in two ways. Since $\boldsymbol{w}_t^*$ is the minimizer of $\tilde{F}_t^*$, we may apply the Fenchel-Young inequality for dual norms to conclude that

$$F_{t+1}(\boldsymbol{w}_t^*, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) = \tilde{F}_t^*(\boldsymbol{w}_t^*) + \langle \boldsymbol{w}_t^*, \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^* \rangle - (\tilde{F}_t^*(\bar{\boldsymbol{w}}_t) + \langle \bar{\boldsymbol{w}}_t, \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^* \rangle) \quad (C.18)$$

$$\leq \langle \boldsymbol{w}_t^* - \bar{\boldsymbol{w}}_t, \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^* \rangle \leq \|\boldsymbol{w}_t^* - \bar{\boldsymbol{w}}_t\| \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_* \leq a\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*. \quad (C.19)$$

Moreover, by [129, proof of Thm. 7.28] and the fact that $\bar{\boldsymbol{w}}_t$ minimizes $F_{t+1}(\cdot, \lambda_t)$ over $\mathbb{W}$,

$$F_{t+1}(\boldsymbol{w}_t^*, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) \leq \frac{\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*^2}{2\lambda_t}. \quad (C.20)$$

Our collective bounds establish that

$$\delta_t(\tilde{\boldsymbol{g}}_t^*) \triangleq F_{t+1}(\boldsymbol{w}_t^*, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}_t^* \rangle \quad (C.21)$$

$$\leq \min(\frac{1}{2\lambda_t}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*^2, a\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*) + \min(\frac{1}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_*, a\|\boldsymbol{g}_t\|_*) \quad (C.22)$$

$$\leq \frac{1}{2\lambda_t}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*^2 + \frac{1}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_*. \quad (C.23)$$

To obtain an interpretable bound on regret, we will minimize the final expression over all convex combinations $\tilde{\boldsymbol{g}}_t^*$ of $\boldsymbol{g}_t$ and $\tilde{\boldsymbol{g}}_t$. The optimal choice is given by

$$\hat{\boldsymbol{g}}_t = \boldsymbol{g}_t + c_*(\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t) \quad \text{for} \quad (C.24)$$

$$c_* \triangleq \min(\tfrac{\|\boldsymbol{g}_t\|_*}{\|\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t\|_*}, 1) = \operatorname*{arg\,min}_{c \leq 1, \tilde{\boldsymbol{g}}_t^* = \boldsymbol{g}_t + c(\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t)} \tfrac{1}{2\lambda_t}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t^*\|_*^2 + \tfrac{1}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_* \quad (C.25)$$

$$= \arg\min_{c \leq 1} \tfrac{c^2}{2\lambda_t}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*^2 + \tfrac{1-c}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\tilde{\boldsymbol{g}}_t - \boldsymbol{g}_t\|_*. \quad (C.26)$$

For this choice, we obtain the bound

$$\begin{aligned}
(\delta_t(\hat{\boldsymbol{g}}_t))_+ &\leq \frac{1}{2\lambda_t}\|\boldsymbol{g}_t - \hat{\boldsymbol{g}}_t\|_*^2 + \frac{1}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\hat{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t\|_* && \text{(C.27)}\\
&= \frac{c_*^2}{2\lambda_t}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*^2 + \frac{1-c_*}{\lambda_t}\|\boldsymbol{g}_t\|_*\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_* && \text{(C.28)}\\
&= \frac{1}{2\lambda_t}\min(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*)^2 + \frac{1}{\lambda_t}\|\boldsymbol{g}_t\|_*(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_* - \|\boldsymbol{g}_t\|_*)_+ && \text{(C.29)}\\
&= \frac{1}{2\lambda_t}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*^2 - (\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_* - \|\boldsymbol{g}_t\|_*)_+^2) && \text{(C.30)}\\
&= \frac{1}{\lambda_t}\text{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*) && \text{(C.31)}
\end{aligned}$$

and therefore

$$\delta_t = \min(\delta_t(\tilde{\boldsymbol{g}}_t), \delta_t(\boldsymbol{g}_t), \delta_t(\hat{\boldsymbol{g}}_t))_+ \leq \min(\frac{1}{\lambda_t}\text{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*), a\min(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t\|_*)). \tag{C.32}$$

Since $\tilde{\boldsymbol{g}}_t^*$ is arbitrary, the advertised regret bounds follow as

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &\leq \inf_{\tilde{\boldsymbol{g}}_1^*,\ldots,\tilde{\boldsymbol{g}}_T^* \in \mathbb{R}^d} \lambda_{T+1}\psi(\boldsymbol{u}) + \sum_{t=1}^{T}\delta_t(\tilde{\boldsymbol{g}}_t^*) && \text{(C.33)}\\
&= \lambda_{T+1}\psi(\boldsymbol{u}) + \sum_{t=1}^{T}\inf_{\tilde{\boldsymbol{g}}_t^* \in \mathbb{R}^d}\delta_t(\tilde{\boldsymbol{g}}_t^*) && \text{(C.34)}\\
&\leq \lambda_{T+1}\psi(\boldsymbol{u}) + \sum_{t=1}^{T}\min(\delta_t(\tilde{\boldsymbol{g}}_t), \delta_t(\boldsymbol{g}_t), \delta_t(\hat{\boldsymbol{g}}_t))_+. && \text{(C.35)}
\end{aligned}$$

■

C.2.1 Proof of Lem. C.2.1: OAFTRL difference bound

Fix any time t, and define the optimistic objective function $\tilde{F}_t(\boldsymbol{w}) = \lambda_t\psi(\boldsymbol{w}) + \sum_{i=1}^{t-1}\langle\boldsymbol{g}_i, \boldsymbol{w}\rangle + \langle\tilde{\boldsymbol{g}}_t, \boldsymbol{w}\rangle$ and the auxiliary optimistic objective function $\tilde{F}_t^*(\boldsymbol{w}) = \lambda_t\psi(\boldsymbol{w}) + \sum_{i=1}^{t-1}\langle\boldsymbol{g}_i, \boldsymbol{w}\rangle +$

$\langle \tilde{\boldsymbol{g}}_t^*, \boldsymbol{w} \rangle$ so that $\boldsymbol{w}_t \in \arg\min_{\boldsymbol{w} \in \mathbb{W}} \tilde{F}_t(\boldsymbol{w})$ and $\boldsymbol{w}_t^* \in \arg\min_{\boldsymbol{w} \in \mathbb{W}} \tilde{F}_t^*(\boldsymbol{w})$. We have

$$\tilde{F}_t^*(\boldsymbol{w}_t) - \tilde{F}_t^*(\boldsymbol{w}_t^*) \geq \frac{\lambda_t}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2 \quad \text{by the strong convexity of } \tilde{F}_t^* \text{ and} \tag{C.36}$$

$$\tilde{F}_t(\boldsymbol{w}_t^*) - \tilde{F}_t(\boldsymbol{w}_t) \geq \frac{\lambda_t}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2 \quad \text{by the strong convexity of } \tilde{F}_t. \tag{C.37}$$

Summing the above inequalities and applying the Fenchel-Young inequality for dual norms, we obtain

$$\lambda_t \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|^2 \leq \langle \tilde{\boldsymbol{g}}_t^* - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t - \boldsymbol{w}_t^* \rangle \leq \|\tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_t^*\|_* \|\boldsymbol{w}_t - \boldsymbol{w}_t^*\|, \tag{C.38}$$

which yields the first half of our target bound after rearrangement. The second half follows from the definition of diameter, as $\|\boldsymbol{w}_t - \boldsymbol{w}_t^*\| \leq \operatorname{diam}(\mathbb{W})$.

C.3 Proof of Thm. 5.4.2: SOOMD regret

We will prove the following more general result for adaptive SOOMD (ASOOMD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{w} \rangle + \lambda_{t+1} \mathcal{B}_\psi(\boldsymbol{w}, \boldsymbol{w}_t) \tag{ASOOMD}$$

$$\text{with arbitrary} \quad \boldsymbol{w}_0 \quad \text{and} \quad \boldsymbol{g}_0 = \tilde{\boldsymbol{g}}_0 = \boldsymbol{0} \tag{C.39}$$

from which Thm. 5.4.2 will follow with the choice $\lambda_t = \lambda$ for all $t \geq 1$.

Theorem C.3.1 (ASOOMD regret)**.** *Fix any $\lambda_{T+1} \geq 0$. If each $(\lambda_{t+1} - \lambda_t)\psi$ is proper and differentiable, $\lambda_0 \triangleq 0$, and $\tilde{\boldsymbol{g}}_{T+1} \triangleq \boldsymbol{0}$, then, for all $\boldsymbol{u} \in \mathbb{W}$, the* ASOOMD *iterates $\boldsymbol{w}_t$ satisfy*

$$\operatorname{Regret}_T(\boldsymbol{u}) \leq \sum_{t=0}^{T} (\lambda_{t+1} - \lambda_t) \mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_t) + \tag{C.40}$$

$$\sum_{t=1}^{T} \min\big(\operatorname{diam}(\mathbb{W}) \|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \frac{1}{\lambda_{t+1}} \operatorname{huber}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \|\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t\|_*)\big). \tag{C.41}$$

Proof. Fix any $\boldsymbol{u} \in \mathbb{W}$, instantiate the notation of [76, Sec. 7.2], and consider the choices

- $r_1 = \lambda_2 \psi$, $r_t = (\lambda_{t+1} - \lambda_t)\psi$ for $t \geq 2$, so that $r_{1:t} = \lambda_{t+1}\psi$ for $t \geq 1$,
- $q_t = \tilde{q}_t + \langle \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \cdot \rangle$ for $t \geq 0$,

- $\tilde{q}_0(\boldsymbol{w}) = \lambda_1 \mathcal{B}_\psi(\boldsymbol{w}, \boldsymbol{w}_0)$ and $\tilde{q}_t \equiv 0$ for all $t \geq 1$,
- $p_1 \triangleq r_1 - q_0 = r_1 - \tilde{q}_0 - \langle \tilde{\boldsymbol{g}}_1 - \tilde{\boldsymbol{g}}_0, \cdot \rangle = \lambda_2 \psi - \lambda_1 \mathcal{B}_\psi(\cdot, \boldsymbol{w}_0) - \langle \tilde{\boldsymbol{g}}_1 - \tilde{\boldsymbol{g}}_0, \cdot \rangle$,
- $p_t \triangleq r_t - q_{t-1} = r_t - \tilde{q}_{t-1} - \langle \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_{t-1}, \cdot \rangle = (\lambda_{t+1} - \lambda_t)\psi - \langle \tilde{\boldsymbol{g}}_t - \tilde{\boldsymbol{g}}_{t-1}, \cdot \rangle$ for all $t \geq 2$.

Since, for each t, $\delta_t = 0$ and ℓ_t is convex, the Ada-MD regret inequality of [76, Eq. (24)] and the choice $\tilde{\boldsymbol{g}}_{T+1} = 0$ imply that

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) && \text{(C.42)}\\
&\leq -\sum_{t=1}^{T} \mathcal{B}_{\ell_t}(\boldsymbol{u}, \boldsymbol{w}_t) + \sum_{t=0}^{T} q_t(\boldsymbol{u}) - q_t(\boldsymbol{w}_{t+1}) + \sum_{t=1}^{T} \mathcal{B}_{p_t}(\boldsymbol{u}, \boldsymbol{w}_t) && \text{(C.43)}\\
&\quad - \sum_{t=1}^{T} \mathcal{B}_{r_{1:t}}(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) + \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1} \rangle + \sum_{t=1}^{T} \delta_t && \text{(C.44)}\\
&\leq \lambda_1(\mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_0) - \mathcal{B}_\psi(\boldsymbol{w}_1, \boldsymbol{w}_0)) + \sum_{t=0}^{T} \langle \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{u} - \boldsymbol{w}_{t+1} \rangle && \text{(C.45)}\\
&\quad + \sum_{t=1}^{T} (\lambda_{t+1} - \lambda_t) \mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_t) + \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1} \rangle - \lambda_{t+1} \mathcal{B}_\psi(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) && \text{(C.46)}\\
&= \sum_{t=0}^{T} (\lambda_{t+1} - \lambda_t) \mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_t) + \sum_{t=0}^{T} \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1} \rangle - \lambda_{t+1} \mathcal{B}_\psi(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t). && \text{(C.47)}
\end{aligned}$$

To obtain our advertised bound, we begin with the expression (C.47) and invoke the 1-strong convexity of ψ and the nonnegativity of $\mathcal{B}_{\lambda\psi}(\boldsymbol{w}_1, \boldsymbol{w}_0)$ to find

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &\leq \sum_{t=0}^{T} (\lambda_{t+1} - \lambda_t) \mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_t) + \sum_{t=0}^{T} \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1} \rangle - \lambda_{t+1} \mathcal{B}_\psi(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) && \text{(C.48)}\\
&\leq \sum_{t=0}^{T} (\lambda_{t+1} - \lambda_t) \mathcal{B}_\psi(\boldsymbol{u}, \boldsymbol{w}_t) + \sum_{t=1}^{T} \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1} \rangle - \frac{\lambda_{t+1}}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\|^2. && \text{(C.49)}
\end{aligned}$$

We will bound the final sum in this expression using two lemmas. The first is a bound on the difference between subsequent ASOOMD iterates distilled from [75, proof of Prop. 2].

Lemma C.3.1 (ASOOMD iterate bound [75, proof of Prop. 2])**.** *If ψ is differentiable and 1-strongly convex with respect to $\|\cdot\|$, then the* ASOOMD *iterates satisfy*

$$\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\| \leq \frac{1}{\lambda_{t+1}} \|\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t\|_*. \tag{C.50}$$

The second, proved in App. C.3.1, is a general bound on $\langle \boldsymbol{g}, \mathbf{v}\rangle - \frac{\lambda}{2}\|\mathbf{v}\|^2$ under a norm constraint on $\mathbf{v}$.

Lemma C.3.2 (Norm-constrained conjugate)**.** *For any $\boldsymbol{g} \in \mathbb{R}^d$ and $\lambda, c, b > 0$,*

$$\begin{aligned}
\sup_{\mathbf{v}\in\mathbb{R}^d : \|\mathbf{v}\| \leq \min(\frac{c}{\lambda}, b)} \langle \boldsymbol{g}, \mathbf{v}\rangle - \tfrac{\lambda}{2}\|\mathbf{v}\|^2 &= \tfrac{1}{\lambda}\min(\|\boldsymbol{g}\|_*, c, b\lambda)(\|\boldsymbol{g}\|_* - \tfrac{1}{2}\min(\|\boldsymbol{g}\|_*, c, b\lambda)) && \text{(C.51)}\\
&\leq \min(b\|\boldsymbol{g}\|_*, \tfrac{1}{\lambda}\min(\|\boldsymbol{g}\|_*, c)(\|\boldsymbol{g}\|_* - \tfrac{1}{2}\min(\|\boldsymbol{g}\|_*, c))) && \text{(C.52)}\\
&= \min(b\|\boldsymbol{g}\|_*, \tfrac{1}{2\lambda}(\|\boldsymbol{g}\|_*^2 - (\|\boldsymbol{g}\|_* - \min(\|\boldsymbol{g}\|_*, c))^2)) && \text{(C.53)}\\
&= \min(b\|\boldsymbol{g}\|_*, \tfrac{1}{2\lambda}(\|\boldsymbol{g}\|_*^2 - (\|\boldsymbol{g}\|_* - c)_+^2)) && \text{(C.54)}\\
&\leq \min(\tfrac{1}{2\lambda}\|\boldsymbol{g}\|_*^2, \tfrac{1}{\lambda}c\|\boldsymbol{g}\|_*, b\|\boldsymbol{g}\|_*). && \text{(C.55)}
\end{aligned}$$

By Lems. C.3.1 and C.3.2 and the definition of $a \triangleq \operatorname{diam}(\mathbb{W})$, each summand in our regret bound (C.49) satisfies

$$\begin{aligned}
&\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t - \boldsymbol{w}_{t+1}\rangle - \tfrac{\lambda_{t+1}}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t+1}\|^2 && \text{(C.56)}\\
&\leq \sup_{\mathbf{v}\in\mathbb{R}^d : \|\mathbf{v}\| \leq \min(\frac{1}{\lambda_{t+1}}\|\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t\|_*, a)} \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \mathbf{v}\rangle - \tfrac{\lambda_{t+1}}{2}\|\mathbf{v}\|^2 && \text{(C.57)}\\
&= \min\left(a\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*, \tfrac{1}{2\lambda_{t+1}}(\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_*^2 - (\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_* - \|\boldsymbol{g}_t + \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t\|_*)_+^2)\right) && \text{(C.58)}
\end{aligned}$$

yielding the advertised result. ■

C.3.1 Proof of Lem. C.3.2: Norm-constrained conjugate

By the definition of the dual norm,

$$\sup_{\mathbf{v}\in\mathbb{R}^d:\|\mathbf{v}\|\leq\min(\frac{c}{\lambda},b)} \langle \boldsymbol{g},\mathbf{v}\rangle - \frac{\lambda}{2}\|\mathbf{v}\|^2 \tag{C.59}$$

$$= \sup_{a\leq\min(\frac{c}{\lambda},b)} \sup_{\mathbf{v}\in\mathbb{R}^d:\|\mathbf{v}\|\leq a} \langle \boldsymbol{g},\mathbf{v}\rangle - \tfrac{\lambda}{2}a^2 = \sup_{a\leq\min(\frac{c}{\lambda},b)} a\|\boldsymbol{g}\|_* - \tfrac{\lambda}{2}a^2 \tag{C.60}$$

$$= \tfrac{1}{\lambda}\min(\|\boldsymbol{g}\|_*, c, b\lambda)(\|\boldsymbol{g}\|_* - \tfrac{1}{2}\min(\|\boldsymbol{g}\|_*, c, b\lambda)) \tag{C.61}$$

$$\leq \min(\frac{1}{\lambda}c\|\boldsymbol{g}\|_*, b\|\boldsymbol{g}\|_*). \tag{C.62}$$

We compare to the values of less constrained optimization problems to obtain the final inequalities:

$$\sup_{a\leq\min(\frac{c}{\lambda},b)} a\|\boldsymbol{g}\|_* - \tfrac{\lambda}{2}a^2 \leq \sup_{a\leq\frac{c}{\lambda}} a\|\boldsymbol{g}\|_* - \tfrac{\lambda}{2}a^2 = \tfrac{1}{\lambda}\min(\|\boldsymbol{g}\|_*, c)(\|\boldsymbol{g}\|_* - \tfrac{1}{2}\min(\|\boldsymbol{g}\|_*, c)) \tag{C.63}$$

$$\leq \sup_{a>0} a\|\boldsymbol{g}\|_* - \tfrac{\lambda}{2}a^2 = \tfrac{1}{\lambda}\tfrac{1}{2}\|\boldsymbol{g}\|_*^2. \tag{C.64}$$

C.4 Proof of Lem. 5.6.2: DORM is ODAFTRL and DORM + is DOOMD

Our derivations will make use of several facts about ℓ^p norms, summarized in the next lemma.

Lemma C.4.1 (ℓ^p norm facts)**.** *For $p \in (1,\infty)$, $\psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_p^2$, and any vectors $\boldsymbol{w}, \mathbf{v} \in \mathbb{R}^d$*

and $\tilde{\boldsymbol{w}}_0 \in \mathbb{R}^d_+$,

$$\nabla\psi(\boldsymbol{w}) = \nabla\tfrac{1}{2}\|\boldsymbol{w}\|_p^2 = \operatorname{sign}(\boldsymbol{w})|\boldsymbol{w}|^{p-1}/\|\boldsymbol{w}\|_p^{p-2} \tag{C.65}$$

$$\langle \boldsymbol{w}, \nabla\psi(\boldsymbol{w})\rangle = \|\boldsymbol{w}\|_p^2 = 2\psi(\boldsymbol{w}) \tag{C.66}$$

$$\psi^*(\mathbf{v}) = \sup_{\boldsymbol{w}\in\mathbb{R}^d} \langle \boldsymbol{w}, \mathbf{v}\rangle - \psi(\boldsymbol{w}) = \tfrac{1}{2}\|\mathbf{v}\|_q^2 \quad \textit{for} \quad 1/q = 1 - 1/p \tag{C.67}$$

$$\nabla\psi^*(\mathbf{v}) = \operatorname{sign}(\mathbf{v})|\mathbf{v}|^{q-1}/\|\mathbf{v}\|_q^{q-2} \tag{C.68}$$

$$\psi^*_+(\mathbf{v}) = \sup_{\boldsymbol{w}\in\mathbb{R}^d_+} \langle \boldsymbol{w}, \mathbf{v}\rangle - \psi(\boldsymbol{w}) \tag{C.69}$$

$$= \sup_{\boldsymbol{w}\in\mathbb{R}^d} \langle \boldsymbol{w}, (\mathbf{v})_+\rangle - \psi(\boldsymbol{w}) = \tfrac{1}{2}\|(\mathbf{v})_+\|_q^2 \tag{C.70}$$

$$\nabla\psi^*_+(\mathbf{v}) = \operatorname*{arg\,max}_{\boldsymbol{w}\in\mathbb{R}^d_+} \langle \boldsymbol{w}, \mathbf{v}\rangle - \psi(\boldsymbol{w}) = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^d_+} \psi(\boldsymbol{w}) - \langle \boldsymbol{w}, \mathbf{v}\rangle \tag{C.71}$$

$$= (\mathbf{v})_+^{q-1}/\|(\mathbf{v})_+\|_q^{q-2} \tag{C.72}$$

$$\min_{\tilde{\boldsymbol{w}}\in\mathbb{R}^d_+} \mathcal{B}_{\lambda\psi}(\tilde{\boldsymbol{w}}, \tilde{\boldsymbol{w}}_0) - \langle \mathbf{v}, \tilde{\boldsymbol{w}}\rangle = \lambda(\langle \tilde{\boldsymbol{w}}_0, \nabla\psi(\tilde{\boldsymbol{w}}_0)\rangle - \psi(\tilde{\boldsymbol{w}}_0) - \sup_{\tilde{\boldsymbol{w}}\in\mathbb{R}^d_+} \langle \tilde{\boldsymbol{w}}, \nabla\psi(\tilde{\boldsymbol{w}}_0) + \mathbf{v}/\lambda\rangle - \psi(\tilde{\boldsymbol{w}})) \tag{C.73}$$

$$= \lambda(\langle \tilde{\boldsymbol{w}}_0, \nabla\psi(\tilde{\boldsymbol{w}}_0)\rangle - \psi(\tilde{\boldsymbol{w}}_0) - \psi^*_+(\nabla\psi(\tilde{\boldsymbol{w}}_0) + \mathbf{v}/\lambda)) \tag{C.74}$$

$$= \lambda(\psi(\tilde{\boldsymbol{w}}_0) - \psi^*_+(\nabla\psi(\tilde{\boldsymbol{w}}_0) + \mathbf{v}/\lambda)) \tag{C.75}$$

$$= \lambda(\psi(\tilde{\boldsymbol{w}}_0) - \tfrac{1}{2}\|(\nabla\psi(\tilde{\boldsymbol{w}}_0) + \mathbf{v}/\lambda)_+\|_q^2) \tag{C.76}$$

$$= \lambda(\tfrac{1}{2}\|\tilde{\boldsymbol{w}}_0\|_p^2 - \tfrac{1}{2}\|(\tilde{\boldsymbol{w}}_0^{p-1}/\|\tilde{\boldsymbol{w}}_0\|_p^{p-2} + \mathbf{v}/\lambda)_+\|_q^2). \tag{C.77}$$

Proof. The fact (C.65) follows from the chain rule as

$$\nabla_j \frac{1}{2}\|\boldsymbol{w}\|_p^2 = \frac{1}{2}\nabla_j(\|\boldsymbol{w}\|_p^p)^{2/p} = \frac{1}{p}(\|\boldsymbol{w}\|_p^p)^{(2/p)-1}\nabla_j\|\boldsymbol{w}\|_p^p = \frac{1}{p}\|\boldsymbol{w}\|_p^{2-p}\nabla_j \sum_{j'=1}^{d} |\boldsymbol{w}_{j'}|^p \tag{C.78}$$

$$= \frac{1}{p}\|\boldsymbol{w}\|_p^{2-p} p \operatorname{sign}(\boldsymbol{w}_j)|\boldsymbol{w}_j|^{p-1} = \operatorname{sign}(\boldsymbol{w}_j)|\boldsymbol{w}_j|^{p-1}/\|\boldsymbol{w}\|_p^{p-2}. \tag{C.79}$$

The fact (C.67) follows from Lem. C.3.2 as $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$. ■

We now prove each claim in turn.

C.4.1 DORM is ODAFTRL

Fix $p \in (1,2]$, $\lambda > 0$, and $t \geq 0$. The ODAFTRL iterate with hint $-\boldsymbol{h}_{t+1}$, $\mathbb{W} \triangleq \mathbb{R}^d_+$, $\psi(\tilde{\boldsymbol{w}}) = \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_p^2$, loss subgradients $\boldsymbol{g}_{1:t-D}^{\text{ODAFTRL}} = -\boldsymbol{r}_{1:t-D}$, and regularization parameter λ takes the form

$$\begin{aligned}
&\operatorname*{arg\,min}_{\tilde{\boldsymbol{w}} \in \mathbb{R}^d_+} \lambda\psi(\tilde{\boldsymbol{w}}) - \langle \tilde{\boldsymbol{w}}, \boldsymbol{h}_{t+1} + \boldsymbol{r}_{1:t-D}\rangle && \text{(C.80)}\\
&\qquad = \operatorname*{arg\,min}_{\tilde{\boldsymbol{w}} \in \mathbb{R}^d_+} \psi(\tilde{\boldsymbol{w}}) - \langle \tilde{\boldsymbol{w}}, (\boldsymbol{h}_{t+1} + \boldsymbol{r}_{1:t-D})/\lambda\rangle && \text{(C.81)}\\
&\qquad = ((\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1}/\|((\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})/\lambda)_+\|_q^{q-2} \quad \text{by (C.72)} && \text{(C.82)}\\
&\qquad = ((\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1}\|((\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1}\|_p^{p-2} && \text{(C.83)}\\
&\qquad\quad \text{since } (p-1)(q-1) = 1\\
&\qquad = \tilde{\boldsymbol{w}}_{t+1}\|\tilde{\boldsymbol{w}}_{t+1}\|_p^{p-2} && \text{(C.84)}
\end{aligned}$$

proving the claim.

C.4.2 DORM+ is DOOMD

Fix $p \in (1,2]$ and $\lambda > 0$, and let $(\tilde{\boldsymbol{w}}_t)_{t\geq 0}$ denote the unnormalized iterates generated by DORM+ with hints $\boldsymbol{h}_t$, instantaneous regrets $\boldsymbol{r}_t$, regularization parameter λ, and hyperparameter q. For $p = q/(q-1)$, let $(\bar{\boldsymbol{w}}_t)_{t\geq 0}$ denote the sequence generated by DOOMD with $\bar{\boldsymbol{w}}_0 = \mathbf{0}$, hints $-\boldsymbol{h}_t$, $\mathbb{W} \triangleq \mathbb{R}^d_+$, $\psi(\tilde{\boldsymbol{w}}) = \frac{1}{2}\|\tilde{\boldsymbol{w}}\|_p^2$, loss subgradients $\boldsymbol{g}_t^{\text{DOOMD}} = -\boldsymbol{r}_t$, and regularization parameter λ. We proceed by induction to show that, for each t, $\bar{\boldsymbol{w}}_t = \tilde{\boldsymbol{w}}_t\|\tilde{\boldsymbol{w}}_t\|_p^{p-2}$.

Base case By assumption, $\bar{\boldsymbol{w}}_0 = \mathbf{0} = \tilde{\boldsymbol{w}}_0\|\tilde{\boldsymbol{w}}_0\|_p^{p-2}$, confirming the base case.

Inductive step Fix any $t \geq 0$ and assume that for each $s \leq t$, $\bar{\boldsymbol{w}}_s = \tilde{\boldsymbol{w}}_s \|\tilde{\boldsymbol{w}}_s\|_p^{p-2}$. Then, by the definition of DOOMD and our ℓ^p norm facts,

$$\bar{\boldsymbol{w}}_{t+1} = \operatorname*{arg\,min}_{\bar{\boldsymbol{w}} \in \mathbb{R}_+^d} \langle -\boldsymbol{h}_{t+1} + \boldsymbol{h}_t - \boldsymbol{r}_{t-D}, \bar{\boldsymbol{w}} \rangle + \mathcal{B}_{\lambda\psi}(\bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}_t) \tag{C.85}$$

$$= \operatorname*{arg\,min}_{\bar{\boldsymbol{w}} \in \mathbb{R}_+^d} \lambda(\psi(\bar{\boldsymbol{w}}) - \psi(\bar{\boldsymbol{w}}_t) - \langle \bar{\boldsymbol{w}} - \bar{\boldsymbol{w}}_t, \nabla\psi(\bar{\boldsymbol{w}}_t) \rangle) + \langle -\boldsymbol{h}_{t+1} + \boldsymbol{h}_t - \boldsymbol{r}_{t-D}, \bar{\boldsymbol{w}} \rangle \tag{C.86}$$

$$= \operatorname*{arg\,min}_{\bar{\boldsymbol{w}} \in \mathbb{R}_+^d} \psi(\bar{\boldsymbol{w}}) - \langle \bar{\boldsymbol{w}}, \nabla\psi(\bar{\boldsymbol{w}}_t) + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda \rangle \tag{C.87}$$

$$= \operatorname*{arg\,min}_{\bar{\boldsymbol{w}} \in \mathbb{R}_+^d} \psi(\bar{\boldsymbol{w}}) - \langle \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}_t^{p-1} / \|\bar{\boldsymbol{w}}_t\|_p^{p-2} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda \rangle \quad \text{by (C.65)} \tag{C.88}$$

$$= \operatorname*{arg\,min}_{\bar{\boldsymbol{w}} \in \mathbb{R}_+^d} \psi(\bar{\boldsymbol{w}}) - \langle \bar{\boldsymbol{w}}, \tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda \rangle \quad \text{by the inductive hypothesis} \tag{C.89}$$

$$= (\tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1} / \|(\tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda)_+\|_q^{q-2} \tag{C.90}$$

by (C.72)

$$= (\tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1} \|(\tilde{\boldsymbol{w}}_t^{p-1} + (\boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})/\lambda)_+^{q-1}\|_p^{p-2} \tag{C.91}$$

since $(p-1)(q-1) = 1$

$$= \tilde{\boldsymbol{w}}_{t+1} \|\tilde{\boldsymbol{w}}_{t+1}\|_p^{p-2}, \tag{C.92}$$

completing the inductive step.

C.5 Proof of Lem. 5.6.1: DORM and DORM+ are independent of λ

We will prove the following more general result, from which the stated result follows immediately.

Lemma C.5.1 (DORM and DORM+ are independent of λ)**.** *Consider either* DORM *or* DORM+ *plays* $\tilde{\boldsymbol{w}}_t$ *as a function of* $\lambda > 0$*, and suppose that for all time points* t*, the observed subgradient* $\boldsymbol{g}_t$ *and chosen hint* $\boldsymbol{h}_{t+1}$ *only depend on* λ *through* $(\boldsymbol{w}_s, \lambda^{q-1}\tilde{\boldsymbol{w}}_s, \boldsymbol{g}_{s-1}, \boldsymbol{h}_s)_{s \leq t}$ *and* $(\boldsymbol{w}_s, \lambda^{q-1}\tilde{\boldsymbol{w}}_s, \boldsymbol{g}_s, \boldsymbol{h}_s)_{s \leq t}$ *respectively. Then if* $\lambda^{q-1}\tilde{\boldsymbol{w}}_0$ *is independent of the choice of* $\lambda > 0$*, then so is* $\lambda^{q-1}\tilde{\boldsymbol{w}}_t$ *for all time points* t*. As a result,* $\boldsymbol{w}_t \propto \lambda^{q-1}\tilde{\boldsymbol{w}}_t$ *is also independent* *of the choice of* $\lambda > 0$ *at all time points.*

Proof. We prove each result by induction on t.

C.5.1 Scaled DORM iterates $\lambda^{q-1}\tilde{\boldsymbol{w}}_t$ are independent of λ

Base case By assumption, $\boldsymbol{h}_1$ is independent of the choice of $\lambda > 0$. Hence $\lambda^{q-1}\tilde{\boldsymbol{w}}_1 = (\boldsymbol{h}_1)_+^{q-1}$ is independent of $\lambda > 0$, confirming the base case.

Inductive step Fix any $t \geq 0$, suppose $\lambda^{q-1}\tilde{\boldsymbol{w}}_s$ is independent of the choice of $\lambda > 0$ for all $s \leq t$, and consider

$$\lambda^{q-1}\tilde{\boldsymbol{w}}_{t+1} = (\boldsymbol{r}_{1:t-D} + \boldsymbol{h}_{t+1})_+^{q-1}. \tag{C.93}$$

Since $\boldsymbol{r}_{1:t-D}$ depends on λ only through $\boldsymbol{w}_s$ and $\boldsymbol{g}_s$ for $s \leq t - D$, our λ dependence assumptions for $(\boldsymbol{g}_s, \boldsymbol{h}_{s+1})_{s \leq t}$; the fact that, for each s, $\boldsymbol{w}_s \propto \lambda^{q-1}\tilde{\boldsymbol{w}}_s$; and our inductive hypothesis together imply that $\lambda^{q-1}\tilde{\boldsymbol{w}}_{t+1}$ is independent of $\lambda > 0$.

C.5.2 Scaled DORM+ iterates $\lambda^{q-1}\tilde{\boldsymbol{w}}_t$ are independent of λ

Base case By assumption, $\lambda^{q-1}\tilde{\boldsymbol{w}}_0$ is independent of the choice of $\lambda > 0$, confirming the base case.

Inductive step Fix any $t \geq 0$ and suppose $\lambda^{q-1}\tilde{\boldsymbol{w}}_s$ is independent of the choice of $\lambda > 0$ for all $s \leq t$. Since $(p-1)(q-1) = 1$,

$$\lambda^{q-1}\tilde{\boldsymbol{w}}_{t+1} = (\lambda\tilde{\boldsymbol{w}}_t^{p-1} + \boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})_+^{q-1} = ((\lambda^{q-1}\tilde{\boldsymbol{w}}_t)^{p-1} + \boldsymbol{r}_{t-D} - \boldsymbol{h}_t + \boldsymbol{h}_{t+1})_+^{q-1}. \tag{C.94}$$

Since $\boldsymbol{r}_{t-D}$ depends on λ only through $\boldsymbol{w}_{t-D}$ and $\boldsymbol{g}_{t-D}$, our λ dependence assumptions for $(\boldsymbol{g}_s, \boldsymbol{h}_{s+1})_{s \leq t}$; the fact that, for each $s \leq t$, $\boldsymbol{w}_s \propto \lambda^{q-1}\tilde{\boldsymbol{w}}_s$; and our inductive hypothesis together imply that $\lambda^{q-1}\tilde{\boldsymbol{w}}_{t+1}$ is independent of $\lambda > 0$. ■

C.6 Proof of Cor. 5.6.1: DORM and DORM+ regret

Fix any $\lambda > 0$ and $\boldsymbol{u} \in \triangle_{d-1}$, consider the unnormalized DORM or DORM+ iterates $\tilde{\boldsymbol{w}}_t$, and define $\bar{\boldsymbol{w}}_t = \tilde{\boldsymbol{w}}_t \|\tilde{\boldsymbol{w}}_t\|_p^{p-2}$ for each t. For either algorithm, we will bound our regret in terms of the surrogate losses

$$\hat{\ell}_t(\tilde{\boldsymbol{w}}) \triangleq -\langle \boldsymbol{r}_t, \tilde{\boldsymbol{w}}\rangle = \langle \boldsymbol{g}_t, \tilde{\boldsymbol{w}}\rangle - \langle \tilde{\boldsymbol{w}}, \mathbf{1}\rangle\langle \boldsymbol{g}_t, \boldsymbol{w}_t\rangle \tag{C.95}$$

defined for $\tilde{\boldsymbol{w}} \in \mathbb{R}^d_+$. Since $\hat{\ell}_t(\boldsymbol{u}) = \langle \boldsymbol{g}_t, \boldsymbol{u} - \boldsymbol{w}_t\rangle$, $\hat{\ell}_t(\bar{\boldsymbol{w}}_t) = 0$, and each ℓ_t is convex, we have

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) \le \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{u}\rangle = \sum_{t=1}^T \hat{\ell}_t(\bar{\boldsymbol{w}}_t) - \hat{\ell}_t(\boldsymbol{u}). \tag{C.96}$$

For DORM, Lem. 5.6.2 implies that $(\bar{\boldsymbol{w}}_t)_{t\ge 1}$ are ODFTRL iterates, so the ODFTRL regret bound (Thm. 5.4.3) and the fact that ψ is 1-strongly convex with respect to $\|\cdot\| = \sqrt{p-1}\|\cdot\|_p$ [see 152, Lemma 17] with $\|\cdot\|_* = \frac{1}{\sqrt{p-1}}\|\cdot\|_q$ imply

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_p^2 + \frac{1}{\lambda(p-1)}\sum_{t=1}^T \boldsymbol{b}_{t,q}. \tag{C.97}$$

Similarly, for DORM+, Lem. 5.6.2 implies that $(\bar{\boldsymbol{w}}_t)_{t\ge 0}$ are DOOMD iterates with $\bar{\boldsymbol{w}}_0 = \mathbf{0}$, so the DOOMD regret bound (Thm. 5.4.4) and the strong convexity of ψ yield

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \mathcal{B}_{\frac{\lambda}{2}\|\cdot\|_p^2}(\boldsymbol{u}, \mathbf{0}) + \frac{1}{\lambda(p-1)}\sum_{t=1}^T \boldsymbol{b}_{t,q} = \frac{\lambda}{2}\|\boldsymbol{u}\|_p^2 + \frac{1}{\lambda(p-1)}\sum_{t=1}^T \boldsymbol{b}_{t,q}. \tag{C.98}$$

Since, by Lem. 5.6.1, the choice of λ does not impact the iterate sequences played by DORM and DORM+, we may take the infimum over $\lambda > 0$ in these regret bounds. The second advertised inequality comes from the identity $\frac{1}{p-1} = q-1$ and the norm equivalence relations $\|\mathbf{v}\|_q \le d^{1/q}\|\mathbf{v}\|_\infty$ and $\|\mathbf{v}\|_p \le \|\mathbf{v}\|_1 = 1$ for $\mathbf{v} \in \mathbb{R}^d$, as shown in Lem. C.6.1 below. The final claim follows as

$$\begin{aligned}\inf_{q'\ge 2} d^{2/q'}(q'-1) = \inf_{q'\ge 2} 2^{2\log_2(d)/q'}(q'-1) &\le 2^{2\log_2(d)/(2\log_2(d))}(2\log_2(d)-1) && \text{(C.99)}\\ &= 2(2\log_2(d)-1) && \text{(C.100)}\end{aligned}$$

since $d > 1$.

Lemma C.6.1 (Equivalence of p-norms). *If $\boldsymbol{x} \in \mathbb{R}^n$ and $q > q' \ge 1$, then $\|\boldsymbol{x}\|_q \le \|\boldsymbol{x}\|_{q'} \le n^{(1/q'-1/q)}\|\boldsymbol{x}\|_q$.*

Proof. To show $\|\mathbf{x}\|_q \le \|\mathbf{x}\|_{q'}$ for $q > q' \ge 1$, suppose without loss of generality that

$\|\mathbf{x}\|_{q'} = 1$. Then, $\|\mathbf{x}\|_q^q = \sum_{i=1}^n |x_i|^q \le \sum_{i=1}^n |x_i|^{q'} = \|\mathbf{x}\|_{q'}^{q'} = 1$. Hence $\|\mathbf{x}\|_q \le 1 = \|\mathbf{x}\|_{q'}$.

For the inequality $\|\mathbf{x}\|_{q'} \le n^{1/q'-1/q}\|\mathbf{x}\|_q$, applying Hölder's inequality yields

$$\|\mathbf{x}\|_{q'}^{q'} = \sum_{i=1}^n 1 \cdot |x_i|^{q'} \le \left(\sum_{i=1}^n 1\right)^{1-\frac{q'}{q}} \left(\sum_{i=1}^n |x_i|^q\right)^{\frac{q'}{q}} = n^{1-\frac{q'}{q}} \|\mathbf{x}\|_q^{q'}, \tag{C.101}$$

so $\|\mathbf{x}\|_{q'} \le n^{1/q'-1/q}\|\mathbf{x}\|_q$. ■

C.7 Proof of Thm. 5.7.1: ODAFTRL regret

Since ODAFTRL is an instance of OAFTRL with $\tilde{\boldsymbol{g}}_{t+1} = \boldsymbol{h}_{t+1} - \sum_{s=t-D+1}^t \boldsymbol{g}_s$, the ODAFTRL result follows immediately from the OAFTRL regret bound, Thm. C.2.1.

C.8 Proof of Thm. 5.7.2: DUB Regret

Fix any $\boldsymbol{u} \in \mathbb{W}$. By Thm. 5.7.1, ODAFTRL admits the regret bound

$$\text{Regret}_T(\boldsymbol{u}) \le \lambda_T \psi(\boldsymbol{u}) + \sum_{t=1}^T \min(\frac{1}{\lambda_t} \boldsymbol{b}_{t,F}, \boldsymbol{a}_{t,F}). \tag{C.102}$$

To control the second term in this bound, we apply the following lemma proved in App. C.8.1.

Lemma C.8.1 (DUB-style tuning bound)**.** *Fix any $\alpha > 0$ and any non-negative sequences $(a_t)_{t=1}^T$, $(b_t)_{t=1}^T$. If*

$$\Delta_{t+1}^* \triangleq 2 \max_{j \le t-D-1} a_{j-D+1:j} + \sqrt{\sum_{i=1}^{t-D} a_i^2 + 2\alpha b_i} \le \alpha\lambda_{t+1} \quad \textit{for each} \quad t \tag{C.103}$$

then

$$\sum_{t=1}^T \min(b_t^2/\lambda_t, a_t) \le \Delta_{T+D+1}^* \le \alpha\lambda_{T+D+1}. \tag{C.104}$$

Since $\lambda_T \le \lambda_{T+D+1}$, the result now follows by setting $a_t = \boldsymbol{a}_{t,F}$ and $b_t = \boldsymbol{b}_{t,F}$, so that

$$\text{Regret}_T(\boldsymbol{u}) \le \lambda_T \psi(\boldsymbol{u}) + \alpha\lambda_{T+D+1} \le (\psi(\boldsymbol{u}) + \alpha)\lambda_{T+D+1}. \tag{C.105}$$

C.8.1 Proof of Lem. C.8.1: DUB-style tuning bound

We prove the claim

$$\Delta_t \triangleq \sum_{i=1}^{t} \min(b_i/\lambda_i, a_i) \le \Delta^*_{t+D+1} \le \alpha\lambda_{t+D+1} \tag{C.106}$$

by induction on t.

Base case For $t \in [D+1]$,

$$\sum_{i=1}^{t} \min(b_i/\lambda_i, a_i) \le a_{1:t-1} + a_t \le 2 \max_{j\le t-1} a_{j-D+1:j} + \sqrt{\sum_{i=1}^{t} a_i^2 + 2\alpha b_i} = \Delta^*_{t+D+1} \le \alpha\lambda_{t+D+1} \tag{C.107}$$

confirming the base case.

Inductive step Now fix any $t+1 \ge D+2$ and suppose that

$$\Delta_i \le \Delta^*_{i+D+1} \le \alpha\lambda_{i+D+1} \tag{C.108}$$

for all $1 \le i \le t$. We apply this inductive hypothesis to deduce that, for each $0 \le i \le t$,

$$\begin{aligned}
\Delta_{i+1}^2 - \Delta_i^2 &= (\Delta_i + \min(b_{i+1}/\lambda_{i+1}, a_{i+1}))^2 - \Delta_i^2 && \text{(C.109)}\\
&= 2\Delta_i \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 && \text{(C.110)}\\
&= 2\Delta_{i-D} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2(\Delta_i - \Delta_{i-D}) \min(b_{i+1}/\lambda_{i+1}, a_{i+1})\\
&+ \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 && \text{(C.111)}\\
&= 2\Delta_{i-D} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2\sum_{j=i-D+1}^{i} \min(b_j/\lambda_j, a_j) \min(b_{i+1}/\lambda_{i+1}, a_{i+1})\\
&+ \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 && \text{(C.112)}\\
&\le 2\alpha\lambda_{i+1} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2a_{i-D+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + a_{i+1}^2 && \text{(C.113)}\\
&\le 2\alpha b_{i+1} + a_{i+1}^2 + 2a_{i-D+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}). && \text{(C.114)}
\end{aligned}$$

Now, we sum this inequality over $i = 0, \ldots, t$, to obtain

$$\begin{aligned}
\Delta_{t+1}^2 &\leq \sum_{i=0}^{t} (2\alpha b_{i+1} + a_{i+1}^2) + 2\sum_{i=0}^{t} a_{i-D+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) && \text{(C.115)}\\
&= \sum_{i=1}^{t+1} (2\alpha b_i + a_i^2) + 2\sum_{i=1}^{t+1} a_{i-D:i-1} \min(b_i/\lambda_i, a_i) && \text{(C.116)}\\
&\leq \sum_{i=1}^{t+1} (a_i^2 + 2\alpha b_i) + 2\max_{j\leq t} a_{j-D+1:j} \sum_{i=1}^{t+1} \min(b_i/\lambda_i, a_i) && \text{(C.117)}\\
&= \sum_{i=1}^{t+1} (a_i^2 + 2\alpha b_i) + 2\Delta_{t+1} \max_{j\leq t} a_{j-D+1:j}. && \text{(C.118)}
\end{aligned}$$

Solving this quadratic inequality and applying the triangle inequality, we have

$$\begin{aligned}
\Delta_{t+1} &\leq \max_{j\leq t} a_{j-D+1:j} + \frac{1}{2}\sqrt{(2\max_{j\leq t} a_{j-D+1:j})^2 + 4\sum_{i=1}^{t+1} a_i^2 + 2\alpha b_i} && \text{(C.119)}\\
&\leq 2\max_{j\leq t} a_{j-D+1:j} + \sqrt{\sum_{i=1}^{t+1} a_i^2 + 2\alpha b_i} = \Delta_{t+D+2}^* \leq \alpha\lambda_{t+D+2}. && \text{(C.120)}
\end{aligned}$$

C.9 Proof of Thm. 5.7.3: AdaHedgeD Regret

Fix any $\boldsymbol{u} \in \mathbb{W}$. Since the AdaHedgeD regularization sequence $(\lambda_t)_{t\geq 1}$ is non-decreasing, Thm. C.2.1 gives the regret bound

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda_T \psi(\boldsymbol{u}) + \sum_{t=1}^{T} \delta_t = \lambda_T \psi(\boldsymbol{u}) + \alpha\lambda_{T+D+1} \leq (\psi(\boldsymbol{u}) + \alpha)\lambda_{T+D+1}, \quad \text{(C.121)}$$

and the proof of Thm. C.2.1 gives the upper estimate (C.32):

$$\delta_t \leq \min\left(\frac{\boldsymbol{b}_{t,F}}{\lambda_t}, \boldsymbol{a}_{t,F}\right) \quad \text{for all} \quad t \in [T]. \quad \text{(C.122)}$$

Hence, it remains to bound λ_{T+D+1}. Since $\lambda_1 = \cdots = \lambda_{D+1} = 0$ and $\alpha(\lambda_{t+1} - \lambda_t) = \delta_{t-D}$ for $t \geq D+1$,

$$\alpha\lambda_{T+D+1}^2 = \sum_{t=1}^{T+D} \alpha(\lambda_{t+1}^2 - \lambda_t^2) = \sum_{t=D+1}^{T+D} \big(\alpha(\lambda_{t+1} - \lambda_t)^2 + 2\alpha(\lambda_{t+1} - \lambda_t)\lambda_t\big) \tag{C.123}$$

$$= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_{t+D}\big) \quad \text{by the definition of } \lambda_{t+1} \tag{C.124}$$

$$= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_t + 2\delta_t(\lambda_{t+D} - \lambda_t)\big) \tag{C.125}$$

$$\leq \sum_{t=1}^{T} \left(\delta_t^2/\alpha + 2\delta_t\lambda_t + 2\delta_t \max_{t\in[T]}(\lambda_{t+D} - \lambda_t)\right) \tag{C.126}$$

$$= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_t\big) + 2\lambda_{T+D+1} \max_{t\in[T]} \delta_{t-D:t-1} \tag{C.127}$$

$$\leq \sum_{t=1}^{T} \big(\boldsymbol{a}_{t,F}^2/\alpha + 2\boldsymbol{b}_{t,F}\big) + 2\lambda_{T+D+1} \max_{t\in[T]} \boldsymbol{a}_{t-D:t-1,F} \quad \text{by (C.122).} \tag{C.128}$$

Solving the above quadratic inequality for λ_{T+D+1} and applying the triangle inequality, we find

$$\alpha\lambda_{T+D+1} \leq \max_{t\in[T]} \boldsymbol{a}_{t-D:t-1,F} + \frac{1}{2}\sqrt{4(\max_{t\in[T]} \boldsymbol{a}_{t-D:t-1,F})^2 + 4\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha\boldsymbol{b}_{t,F}} \tag{C.129}$$

$$\leq 2\max_{t\in[T]} \boldsymbol{a}_{t-D:t-1,F} + \sqrt{\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha\boldsymbol{b}_{t,F}}. \tag{C.130}$$

C.10 Proof of Thm. 5.8.2: Learning to hint regret

We begin by bounding the hinting problem regret. Since DORM+ is used for the hinting problem, the following result is an immediate corollary of Cor. 5.6.1.

Corollary C.10.1 (DORM+ hinting problem regret)**.** *With convex losses* $l_t(\omega) = f_t(H_t\omega)$

and no meta-hints, the DORM+ *hinting problem iterates* ω_t *satisfy, for each* $v \in \triangle_{m-1}$,

$$\mathrm{HintRegret}_T(v) \triangleq \sum_{t=1}^T l_t(\omega_t) - \sum_{t=1}^T l_t(v) \le \sqrt{\frac{m^{2/q}(q-1)}{2} \sum_{t=1}^T \beta_{t,\infty}} \quad \textit{for} \tag{C.131}$$

$$\beta_{t,\infty} = \begin{cases} \mathrm{huber}(\|\sum_{s=t-D}^t \rho_s\|_\infty, \|\rho_{t-D}\|_\infty), & \textit{for } t < T \\ \frac{1}{2}\|\sum_{s=t-D}^t \rho_s\|_\infty^2, & \textit{for } t = T \end{cases} \tag{C.132}$$

$$\textit{where} \quad \rho_t \triangleq \mathbf{1}\langle \gamma_t, \omega_t \rangle - \gamma_t \quad \textit{for} \quad \gamma_t \in \partial l_t(\omega_t) \tag{C.133}$$

is the instantaneous hinting problem regret.

If, in addition, $q = \arg\min_{q' \ge 2} m^{2/q'}(q'-1)$, *then* $\mathrm{HintRegret}_T(v) \le \sqrt{(2\log_2(m) - 1)\sum_{t=1}^T \beta_{t,\infty}}$.

Our next lemma, proved in App. C.10.1, provides an interpretable bound for each $\beta_{t,\infty}$ term in terms of the hinting problem subgradients $(\gamma_t)_{t \ge 1}$.

Lemma C.10.1 (Hinting problem subgradient regret bound)**.** *Under the notation and assumptions of Cor. C.10.1,*

$$\beta_{t,\infty} \le \begin{cases} \mathrm{huber}(\xi_t, \zeta_t) & \textit{if } t < T \\ \frac{1}{2}\xi_t & \textit{if } t = T \end{cases}, \quad \textit{for} \tag{C.134}$$

$$\xi_t \triangleq 4(D+1) \sum_{s=t-D}^t \|\gamma_s\|_\infty^2 \quad \textit{and} \tag{C.135}$$

$$\zeta_t \triangleq 4\|\gamma_{t-D}\|_\infty \sum_{s=t-D}^t \|\gamma_s\|_\infty. \tag{C.136}$$

Now fix any $\boldsymbol{u} \in \mathbb{W}$. We invoke Assump. 5.8.1, Cor. C.10.1, and Lem. C.10.1 in turn to bound the base problem regret

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u}) && \text{(C.137)}\\
&\le C_0(\boldsymbol{u}) + C_1(\boldsymbol{u})\sqrt{\sum_{t=1}^{T} f_t(\boldsymbol{h}_t(\omega_t))} \quad \text{by Assump. 5.8.1} && \text{(C.138)}\\
&\le C_0(\boldsymbol{u}) + C_1(\boldsymbol{u})\sqrt{\inf_{v\in\mathbb{V}} \sum_{t=1}^{T} f_t(\boldsymbol{h}_t(v)) + \sqrt{(2\log_2(m)-1)\sum_{t=1}^{T}\beta_{t,\infty}}} \quad \text{by Cor. C.10.1} && \text{(C.139)}\\
&\le C_0(\boldsymbol{u}) + C_1(\boldsymbol{u})\sqrt{\inf_{v\in\mathbb{V}} \sum_{t=1}^{T} f_t(\boldsymbol{h}_t(v)) + \sqrt{(2\log_2(m)-1)(\frac{1}{2}\xi_T + \sum_{t=1}^{T-1}\text{huber}(\xi_t,\zeta_t))}} && \text{(C.140)}
\end{aligned}$$

by Lem. C.10.1.

The advertised bound now follows from the triangle inequality.

C.10.1 Proof of Lem. C.10.1: Hinting problem subgradient regret bound

Fix any $t \in [T]$. The triangle inequality implies that

$$\|\rho_t\|_\infty = \|\gamma_t - \mathbf{1}\langle\omega_t,\gamma_t\rangle\|_\infty \le \|\gamma_t\|_\infty + |\langle\omega_t,\gamma_t\rangle| \le 2\|\gamma_t\|_\infty \qquad \text{(C.141)}$$

since $\omega_t \in \triangle_{m-1}$. We repeatedly apply this finding in conjunction with Jensen's inequality to conclude

$$\| \sum_{s=t-D}^{t} \rho_s\|_\infty^2 \le (D+1) \sum_{s=t-D}^{t} \|\rho_s\|_\infty^2 \le 4(D+1) \sum_{s=t-D}^{t} \|\gamma_s\|_\infty^2 \quad \text{and} \qquad \text{(C.142)}$$

$$\|\rho_{t-D}\|_\infty \| \sum_{s=t-D}^{t} \rho_s\|_\infty \le \|\rho_{t-D}\|_\infty \sum_{s=t-D}^{t} \|\rho_s\|_\infty \le 4\|\gamma_{t-D}\|_\infty \sum_{s=t-D}^{t} \|\gamma_s\|_\infty. \qquad \text{(C.143)}$$

C.11 Examples: Learning to Hint with DORM+ and AdaHedgeD

By Thm. 5.7.3, AdaHedgeD satisfies Assump. 5.8.1 with $f_t(\boldsymbol{h}_t) = \|\boldsymbol{r}_t\|_* \|\boldsymbol{h}_t - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_* \geq \frac{\boldsymbol{a}_{t,F}^2 + 2\alpha \boldsymbol{b}_{t,F}}{\operatorname{diam}(\mathbb{W})^2 + 2\alpha}$, $C_1(\boldsymbol{u}) = \sqrt{\operatorname{diam}(\mathbb{W})^2 + 2\alpha}$, and $C_0(\boldsymbol{u}) = 2 \operatorname{diam}(\mathbb{W}) \max_{t \in [T]} \sum_{s=t-D}^{t-1} \|\boldsymbol{g}_s\|_*$.

By Cor. 5.6.1, DORM+ satisfies Assump. 5.8.1 with $f_t(\boldsymbol{h}) = \|\boldsymbol{r}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_q \|\boldsymbol{h} - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_q$, $C_0(\boldsymbol{u}) = 0$, and $C_1(\boldsymbol{u}) = \sqrt{\frac{\|\boldsymbol{u}\|_p^2}{2(p-1)}}$.

These choices give rise to the hinting losses

$$l_t^{\text{DORM+}}(\omega) = \|\boldsymbol{r}_{t-D} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_q \|H_t\omega - \sum_{s=t-D}^{t} \boldsymbol{r}_s\|_q \quad \text{and} \tag{C.144}$$

$$l_t^{\text{AdaHedgeD}}(\omega) = \|\boldsymbol{g}_t\|_q \|H_t\omega - \sum_{s=t-D}^{t} \boldsymbol{g}_s\|_q \quad \text{when} \quad \|\cdot\|_* = \|\cdot\|_q \quad \text{for} \quad q \in [1, \infty]. \tag{C.145}$$

The following lemma, proved in App. C.11.1, identifies subgradients of these hinting losses.

Lemma C.11.1 (Hinting loss subgradient)**.** *If $l_t(\omega) = \|\bar{\boldsymbol{g}}_t\|_q \|H_t\omega - \mathbf{v}_t\|_q$ for some $\bar{\boldsymbol{g}}_t, \mathbf{v}_t \in \mathbb{R}^d$ and $H_t \in \mathbb{R}^{d\times m}$, then*

$$\gamma_t = \begin{cases} \frac{\|\bar{\boldsymbol{g}}_t\|_q}{\|H_t\omega - \mathbf{v}_t\|_q^{q-1}} H_t^\top |H_t\omega - \mathbf{v}_t|^{q-1} \operatorname{sign}(H_t\omega - \mathbf{v}_t) & \text{if } q < \infty \\ \|\bar{\boldsymbol{g}}_t\|_\infty \operatorname{sign}(\mu) H_t^\top \mathbf{e}_k & \text{if } q = \infty \end{cases} \in \quad \partial l_t(\omega) \tag{C.146}$$

for $k = \arg\max_{j\in[d]} (H_t\omega - \mathbf{v}_t)_j$ and $\mu = \max_{j \in [d]} (H_t\omega - \mathbf{v}_t)_j$.

Our next lemma, proved in App. C.11.2, bounds the ∞-norm of this hinting loss subgradient in terms of the base problem subgradients.

Lemma C.11.2 (Hinting loss subgradient bound)**.** *Under the assumptions and notation of Lem. C.11.1, the subgradient γ_t satisfies $\|\gamma_t\|_\infty \leq d^{1/q} \|\bar{\boldsymbol{g}}_t\|_q \|H_t\|_\infty$ for $\|H_t\|_\infty$ the maximum absolute entry of H_t.*

C.11.1 Proof of Lem. C.11.1: Hinting loss subgradient

The result follows immediately from the chain rule and the following lemma.

Lemma C.11.3 (Subgradients of p-norms)**.** *Suppose $\boldsymbol{w} \in \mathbb{R}^d$ and $k \in \arg\max_{j\in[d]} |\boldsymbol{w}_j|$.*

Then

$$\partial\|\boldsymbol{w}\|_p \ni \begin{cases} \frac{|\boldsymbol{w}|^{p-1}}{\|\boldsymbol{w}\|_p^{p-1}}\mathrm{sign}(\boldsymbol{w}) & \textit{if } \|\boldsymbol{w}\|_p \neq 0, p \in [1,\infty) \\ \mathbf{e}_k\mathrm{sign}(\boldsymbol{w}_k) & \textit{if } \|\boldsymbol{w}\|_p \neq 0, p = \infty \\ \mathbf{0} & \textit{if } \|\boldsymbol{w}\|_p = 0 \end{cases} . \tag{C.147}$$

Proof. Since $\mathbf{0}$ is a minimizer of $\|\cdot\|_p$, we have $\|\boldsymbol{u}\|_p \geq \|\mathbf{0}\|_p + \langle \mathbf{0}, \boldsymbol{u} - \mathbf{0}\rangle$ for any $\boldsymbol{u} \in \mathbb{R}^d$ and hence $\mathbf{0} \in \partial\|\mathbf{0}\|_p$.

For $p \in [1, \infty)$, by the chain rule, if $\|\boldsymbol{w}\|_p \neq \mathbf{0}$,

$$\begin{aligned} \partial_j\|\boldsymbol{w}\|_p &= \partial_j\big(\sum_{k=1}^n |\boldsymbol{w}_k|^p\big)^{1/p} = \frac{1}{p}\big(\sum_{k=1}^n |\boldsymbol{w}_k|^p\big)^{(1/p)-1} p|\boldsymbol{w}_j|^{p-1}\mathrm{sign}(\boldsymbol{w}_j) & \text{(C.148)} \\ &= \Big(\big(\sum_{k=1}^n |\boldsymbol{w}_k|^p\big)^{1/p}\Big)^{-(p-1)} |\boldsymbol{w}_j|^{p-1}\mathrm{sign}(\boldsymbol{w}_j) & \text{(C.149)} \\ &= \Big(\frac{|\boldsymbol{w}_j|}{\|\boldsymbol{w}\|_p}\Big)^{p-1}\mathrm{sign}(\boldsymbol{w}_j). & \text{(C.150)} \end{aligned}$$

For $p = \infty$, we have that $\|\boldsymbol{w}\|_\infty = \max_{j\in[n]} |\boldsymbol{w}_j|$. By the Danskin-Bertsekas Theorem [38] for subdifferentials, $\partial\|\boldsymbol{w}\|_\infty = \mathrm{conv}\{\cup\partial|\boldsymbol{w}_j| \quad \text{s.t.} \quad |\boldsymbol{w}_j| = \|\boldsymbol{w}\|_\infty\} = \mathrm{conv}\{\cup\mathrm{sign}(\boldsymbol{w}_j)\mathbf{e}_j \quad \text{s.t.} \quad |\boldsymbol{w}_j| = \|\boldsymbol{w}\|_\infty\}$, where conv is the convex hull operation. ■

C.11.2 Proof of Lem. C.11.2: Hinting loss subgradient bound

If $q \in [1, \infty)$, we have

$$\begin{aligned} \|\gamma_t\|_\infty &= \left\| \frac{\|\bar{\boldsymbol{g}}_t\|_q}{\|H_t\omega - \sum_{s=t-D}^t \boldsymbol{g}_s\|_q^{q-1}} H_t^\top |H_t\omega - \sum_{s=t-D}^t \boldsymbol{g}_s|^{q-1}\mathrm{sign}(H_t\omega - \sum_{s=t-D}^t \boldsymbol{g}_s) \right\|_\infty & \text{(C.151)} \\ &\leq \frac{\|\bar{\boldsymbol{g}}_t\|_q \max_{j\in[d]} \|H_t\mathbf{e}_j\|_q}{\|H_t\omega - \sum_{s=t-D}^t \boldsymbol{g}_s\|_q^{q-1}} \|H_t\omega - \sum_{s=t-D}^t \boldsymbol{g}_s\|_q^{q-1} \quad \text{by Hölder's inequality for } (q,p) & \text{(C.152)} \\ &\leq d^{1/q}\|\bar{\boldsymbol{g}}_t\|_q\|H_t\|_\infty \quad \text{by Lem. C.6.1.} & \text{(C.153)} \end{aligned}$$

If $q = \infty$, we have

$$\|\gamma_t\|_\infty = \left\| \|\bar{\boldsymbol{g}}_t\|_\infty \text{sign}(\mu) H_t^\top \mathbf{e}_k \right\|_\infty = \mathbb{I}[\mu \neq 0] \|\bar{\boldsymbol{g}}_t\|_\infty \|H_t\|_\infty \leq d^{1/q} \|\bar{\boldsymbol{g}}_t\|_\infty \|H_t\|_\infty. \quad \text{(C.154)}$$

C.12 Extended Experimental Results

We present complete experimental results for the four experiments presented in the main text (see Sec. 5.9).

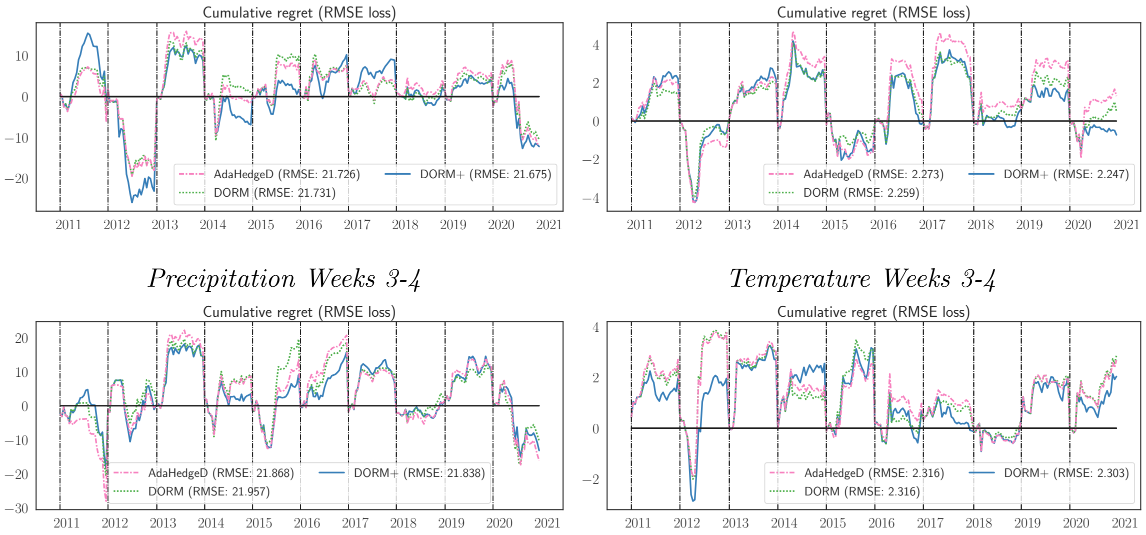
C.12.1 Competing with the Best Input Model

Results for our three delayed online learning algorithms — DORM, DORM+, and AdaHedgeD— on the four subseasonal prediction tasks for the four optimism strategies described in Sec. 5.9 (`recent_g`, `prev_g`, `mean_g`, `none`) are presented below. Each table and figure shows the average RMSE loss and the annual regret versus the best input model in any given year respectively for each algorithm and task.

DORM+ is a competitive model for all three hinting strategies and under the `recent_g` hinting strategy achieves negative regret on all tasks except Temp. 5-6w. For the Temp. 5-6w task, no online learning model outperforms the best input model for any hinting strategy. For the precipitation tasks, the online learning algorithms presented achieve negative regret using all three hinting strategies for all four tasks. Within the subseasonal forecasting domain, precipitation is often considered a more challenging forecasting task than temperature [190]. The gap between the best model and the worst model tends to be larger for precipitation than for temperature, and this could in part explain the strength of the online learning algorithms for these tasks.

*Table C.1: **Hint `recent_g`:** Average RMSE of the 2011-2020 semimonthly forecasts for online learning algorithms (left) and input models (right) over a 10-year evaluation period with the top-performing learners and input models bolded and blue. In each task, the online learners compare favorably with the best input model and learn to downweigh the lower-performing candidates, like the worst model.*

recent_g	AdaHedgeD	DORM	DORM+	Best Input Model	Worst Input
Precip. 3-4w	21.726	21.731	**21.675**	**21.973**	*23.344*
Precip. 5-6w	21.868	21.957	**21.838**	**21.993**	*23.257*
Temp. 3-4w	2.273	2.259	**2.247**	**2.253**	*2.508*
Temp. 5-6w	2.316	2.316	**2.303**	**2.270**	*2.569*



Precipitation Weeks 3-4

Temperature Weeks 3-4

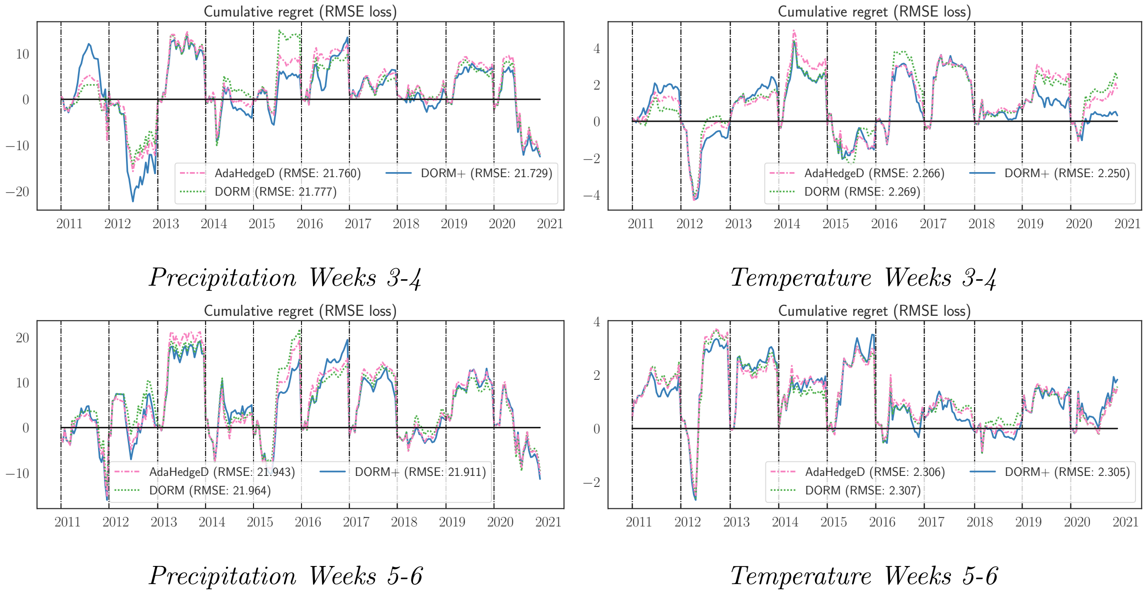
Precipitation Weeks 5-6

Temperature Weeks 5-6

Figure C.1: ***Hint `recent_g`:*** *Yearly cumulative regret under RMSE loss for the three delayed online learning algorithms presented, over the* 10*-year evaluation period. The zero line corresponds to the performance of the best input model in a given year.*

Table C.2: ***Hint `prev_g`:*** *Average RMSE of the 2010-2020 semimonthly forecasts for all four tasks over a* 10*-year evaluation period.*

`prev_g`	AdaHedgeD	DORM	DORM+	Best Input Model	Worst Input
Precip. 3-4w	21.760	21.777	**21.729**	**21.973**	*23.344*
Precip. 5-6w	21.943	21.964	**21.911**	**21.993**	*23.257*
Temp. 3-4w	2.266	2.269	**2.250**	**2.253**	*2.508*
Temp. 5-6w	2.306	2.307	**2.305**	**2.270**	*2.569*



Precipitation Weeks 3-4

Temperature Weeks 3-4

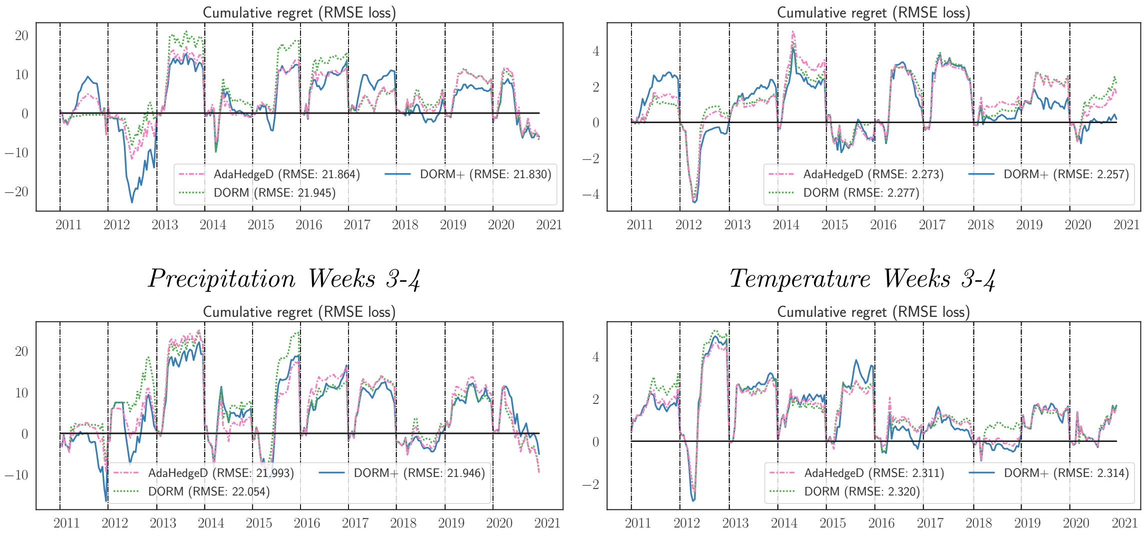
Precipitation Weeks 5-6

Temperature Weeks 5-6

Figure C.2: ***Hint*** **`prev_g`*****:*** *Yearly cumulative regret under RMSE loss for the three delayed online learning algorithms presented.*

Table C.3: ***Hint*** **`mean_g`*****:*** *Average RMSE of the 2010-2020 semimonthly forecasts for all four tasks over a 10-year evaluation period.*

mean_g	AdaHedgeD	DORM	DORM+	Best Input Model	Worst Input
Precip. 3-4w	21.864	21.945	**21.830**	**21.973**	*23.344*
Precip. 5-6w	21.993	22.054	**21.946**	**21.993**	*23.257*
Temp. 3-4w	2.273	2.277	**2.257**	**2.253**	*2.508*
Temp. 5-6w	**2.311**	2.320	2.314	**2.270**	*2.569*



Precipitation Weeks 3-4

Temperature Weeks 3-4

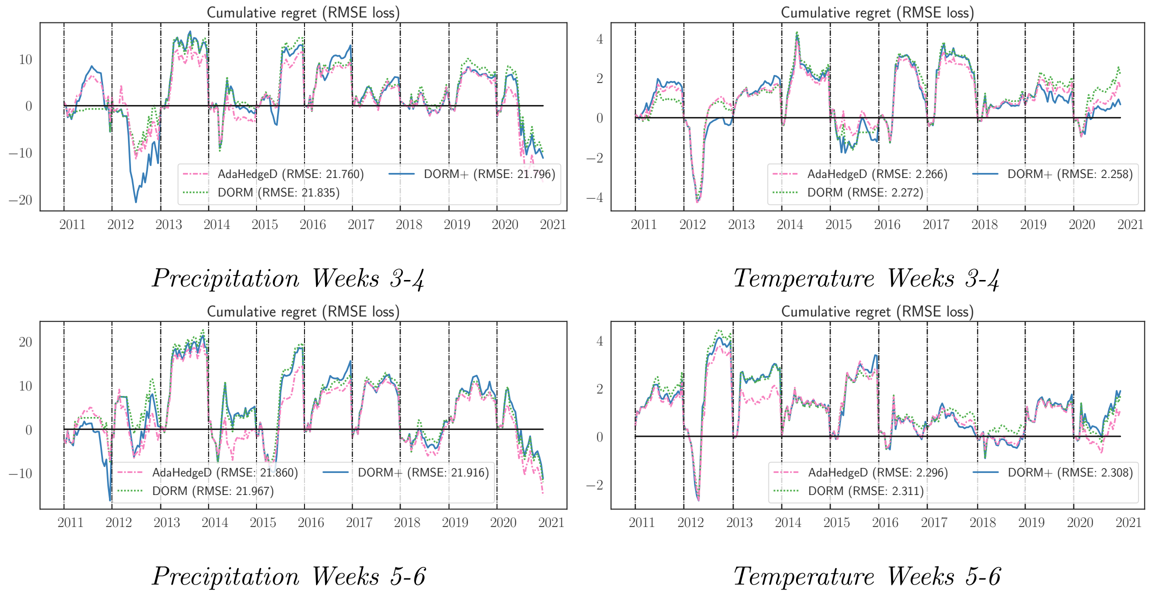
Precipitation Weeks 5-6

Temperature Weeks 5-6

Figure C.3: ***Hint*** **`mean_g`*:*** *Yearly cumulative regret under RMSE loss for the three delayed online learning algorithms presented.*

Table C.4: ***Hint*** **`none`*:*** *Average RMSE of the 2010-2020 semimonthly forecasts for all four tasks over a* 10*-year evaluation period.*

None	AdaHedgeD	DORM	DORM+	Best Input Model	Worst Input
Precip. 3-4w	**21.760**	21.835	21.796	**21.973**	*23.344*
Precip. 5-6w	**21.860**	21.967	21.916	**21.993**	*23.257*
Temp. 3-4w	2.266	2.272	**2.258**	**2.253**	*2.508*
Temp. 5-6w	**2.296**	2.311	2.308	**2.270**	*2.569*



Precipitation Weeks 3-4

Temperature Weeks 3-4

Precipitation Weeks 5-6

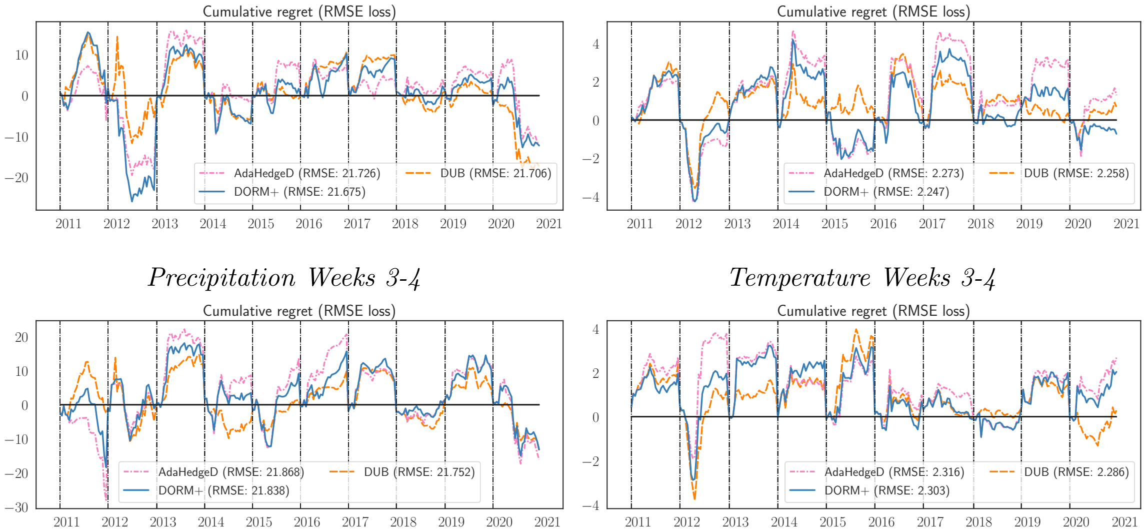
Temperature Weeks 5-6

*Figure C.4: **Hint** `none`: Yearly cumulative regret under RMSE loss for the three delayed online learning algorithms presented.*

C.12.2 Impact of Regularization

Results for three regularization strategies—AdaHedgeD, DORM+, and DUB—on all four subseasonal prediction as described in Sec. 5.9. Fig. C.5 shows the annual regret versus the best input model in any given year for each algorithm and task, and Fig. C.6 presents an example of the weights played by each algorithm in the final evaluation year, as well as the regularization weight used by each algorithm.

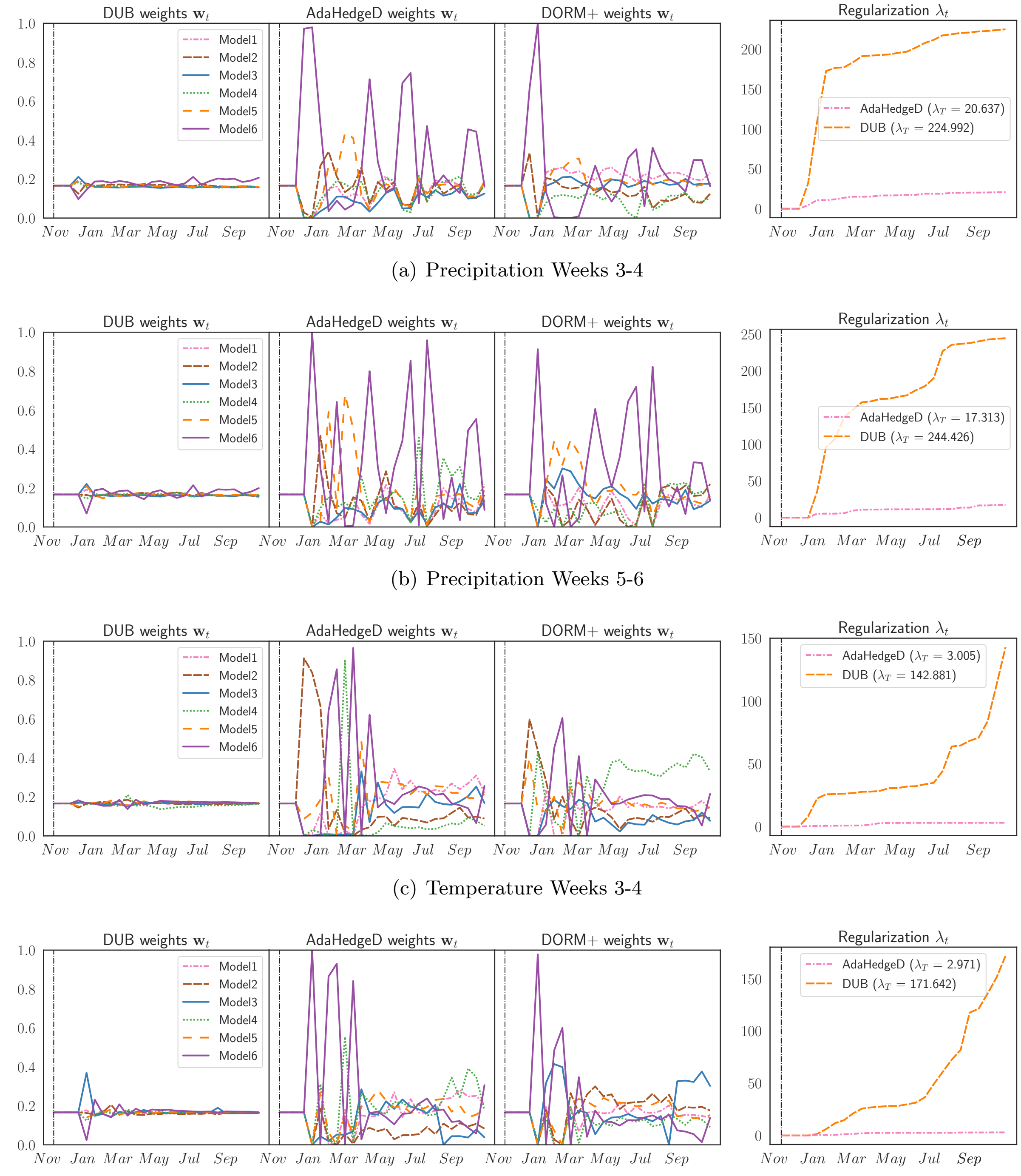
The under- and over-regularization of AdaHedgeD and DUB respectively compared with DORM+ is evident in all four tasks, both in the regret and weight plots. Due to the looseness of the regularization settings used in DUB, its plays can be seen to be very close to the uniform ensemble in all four tasks. For this subseasonal prediction problem, the uniform ensemble is competitive, especially for the 5-6 week horizons. However, in problems where the uniform ensemble has higher regret, this over-regularization property of DUB would be undesirable. The more adaptive plays of DORM+ and AdaHedgeD have the potential to better exploit heterogeneous performance among different input models.



Precipitation Weeks 3-4 *Temperature Weeks 3-4*

Precipitation Weeks 5-6 *Temperature Weeks 5-6*

Figure C.5: ***Overall Regret:*** *Yearly cumulative regret under the RMSE loss for the three regularization algorithms presented.*



(a) Precipitation Weeks 3-4

(b) Precipitation Weeks 5-6

(c) Temperature Weeks 3-4

(d) Temperature Weeks 5-6

Figure C.6: ***Impact of Regularization:*** *The plays* $\boldsymbol{w}_t$ *of online learning algorithms used to combine the input models for all four tasks in the 2020 evaluation year. The weights of* DUB *and* AdaHedgeD *appear respectively over and under regularized compared to* DORM+ *due to their selection of regularization strength* λ_t *(right).*

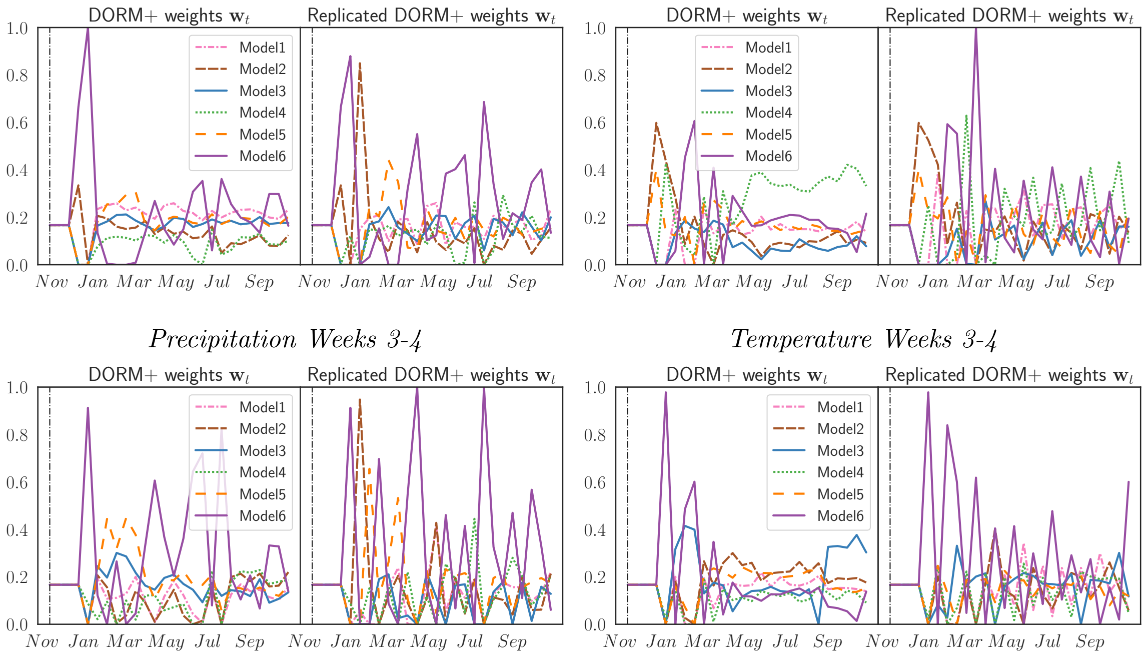
C.12.3 To Replicate or Not to Replicate

We compare the performance of replicated and non-replicated variants of our DORM+ algorithm as in Sec. 5.9. Both algorithms perform well, but in all tasks, DORM+ outperforms replicated DORM+ (in which $D + 1$ independent copies of DORM+ make staggered predictions). Fig. C.7 provides an example of the weight plots produced by the replication strategy for all for tasks.

The replicated algorithms only have the opportunity to learn from $T/(D+1)$ plays. For the 3-4 week horizons tasks $D = 2$ and for the 5-6 week horizons tasks $D = 3$. Because our forecasting horizons are short ($T = 26$), further limiting the feedback available to each online learner via replication could be detrimental to practical model performance.

Table C.5: ***Replication RMSE:*** *Average RMSE of the 2010-2020 semimonthly forecasts for four tasks over a* 10*-year evaluation period for replicated versus standard* DORM+*.*

	DORM+	Replicated DORM+	Best Input Model	Worst Input
Precip. 3-4w	**21.675**	21.720	**21.973**	*23.344*
Precip. 5-6w	**21.838**	21.851	**21.993**	*23.257*
Temp. 3-4w	**2.247**	2.249	**2.253**	*2.508*
Temp. 5-6w	**2.303**	2.315	**2.270**	*2.569*



Precipitation Weeks 3-4

Temperature Weeks 3-4

Precipitation Weeks 5-6

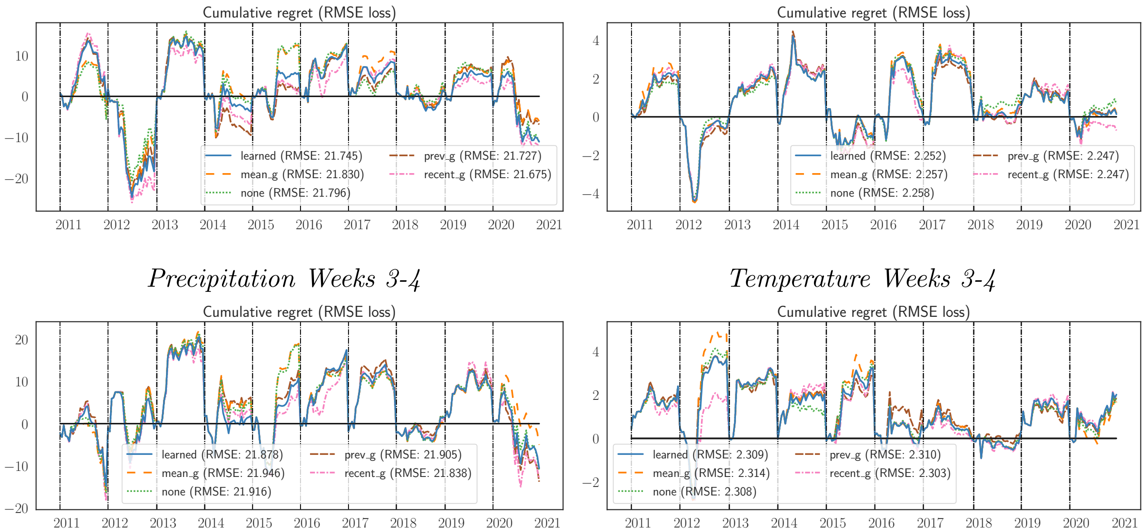
Temperature Weeks 5-6

Figure C.7: ***Replication Weights****: The plays* $\mathbf{w}_t$ *of* DORM+ *and replicated* DORM+ *for all four tasks in the final evaluation year.*

C.12.4 Learning to Hint

We examine the effect of optimism on the DORM+ algorithms and the ability of our "learning to hint" strategy to recover the performance of the best optimism strategy in retrospect as described in Sec. 5.9. We use DORM+ as the meta-algorithm for hint learning to produce the `learned` optimism strategy that plays a convex combination of the three constant hinters.

As reported in the main text, the regret of the base algorithm using the learned hinting strategy generally falls between the worst and the best hinting strategy for any given year. Because the best hinting strategy for any given year is unknown *a priori*, the adaptivity of the hint learner is useful practically. Currently, the hint learner is only optimizing a loose upper bound on base problem regret. Deriving loss functions for hint learning that more accurately quantify the effect of the hinter on base model regret is an important next step in achieving negative regret for online hinting algorithms.



Precipitation Weeks 3-4 *Temperature Weeks 3-4*

Precipitation Weeks 5-6 *Temperature Weeks 5-6*

Figure C.8: ***Overall Regret:*** *Yearly cumulative regret under the RMSE loss for* DORM+ *using the three constant hinting strategies presented and the learned hinter, over the* 10*-year evaluation period.*

C.12.5 Impact of Different Forms of Optimism

The regret analysis presented in this work suggest that optimistic strategies under delay can benefit from hinting at both the "past" $\boldsymbol{g}_{t-D:t-1}$ missing losses and the "future" unobserved loss $\boldsymbol{g}_t$. To study the impact of different forms of optimism on DORM+, we provide a `recent_g` hint for either only the missing **future** loss $\boldsymbol{g}_t$, only the missing **past** losses $\boldsymbol{g}_{t-D:t-1}$, or both **past and future** losses (the strategy used in this thesis) $\boldsymbol{g}_{t-D:t}$. Inspired by the recommendation of an anonymous reviewer, we also test two hint settings that only hint at the future unobserved loss but multiply the weight of that hint by **2D+1** or **3D+1**, effectively increasing the importance of the future hint in the online learning optimization. Fig. C.9 presents the experimental results.

In this experiment, all settings of optimism improve upon the non-optimistic algorithm, and, for all tasks, providing hints for missing future losses outperforms hinting at missing past losses. For all tasks save Temp. 5-6w, hinting at both missing past and future losses yields a further improvement. The 2D+1 and 3D+1 settings demonstrate that, for some tasks, increasing the magnitude of the optimistic hint can further improve performance in line with the online gradient descent predictions of Hsieh et al. [66, Thm. 13].

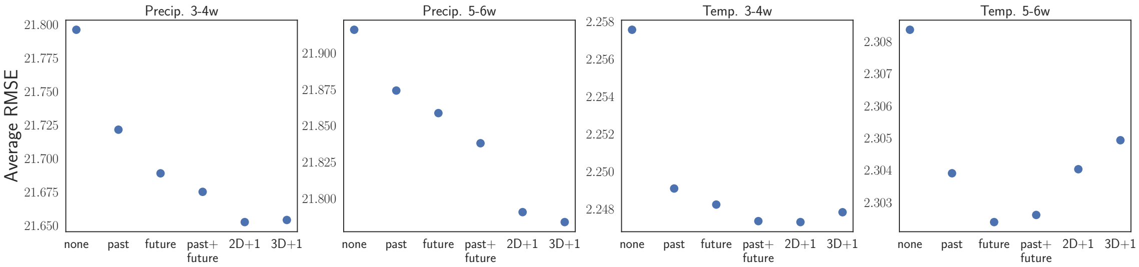


Figure C.9: ***Hinting Regret:*** *DORM+ average RMSE as in Table 5.1 as a function of optimism strategy; see App. C.12.5 for details.*

C.13 Algorithmic Details

C.13.1 ODAFTRL with AdaHedgeD and DUB tuning

The AdaHedgeD and DUB algorithms presented in the experiments are implementations of ODAFTRL with a negative entropy regularizer $\psi(\boldsymbol{w}) = \sum_{j=1}^{d} \boldsymbol{w}_j \ln \boldsymbol{w}_j + \ln d$, which is 1-strongly convex with respect to the norm $\|\cdot\|_1$ [152, Lemma 16] with dual norm $\|\cdot\|_\infty$. Each algorithm optimizes over the simplex and competes with the simplex: $\mathbb{W} = \mathbb{U} = \triangle_{d-1}$. We choose $\alpha = \sup_{\boldsymbol{u} \in \mathbb{U}} \psi(\boldsymbol{u}) = \ln(d)$. In the following, define $\psi_t \triangleq \lambda_t \psi$ for $\lambda_t \geq 0$. Our derivations of the update equations for AdaHedgeD and DUB make use of the following properties of the negative entropy regularizer, proved in App. C.13.4.

Lemma C.13.1 (Negative entropy properties)**.** *The negative entropy regularizer $\psi(\boldsymbol{w}) = \sum_{j=1}^{d} \boldsymbol{w}_j \ln \boldsymbol{w}_j + \ln d$ with $\psi_t = \lambda_t \psi$ for $\lambda_t \geq 0$ satisfies the following properties on the* *simplex* $\mathbb{W} = \triangle_{d-1}$.

$$\psi^*_{\mathbb{W}}(\theta) \triangleq \sup_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{w}, \theta\rangle - \psi(\boldsymbol{w}) = \ln\Big(\sum_{j=1}^{d} \exp(\theta_j)\Big) - \ln d, \tag{C.155}$$

$$(\lambda\psi)^*_{\mathbb{W}}(\theta) \triangleq \sup_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{w}, \theta\rangle - \lambda\psi(\boldsymbol{w}) = \begin{cases} \lambda\psi^*_{\mathbb{W}}(\theta/\lambda) = \lambda\ln(\sum_{j=1}^{d}\exp(\theta_j/\lambda)) - \lambda\ln d, & \textit{if } \lambda > 0 \\ \max_{j\in[d]} \theta_j & \textit{if } \lambda = 0 \end{cases}, \tag{C.156}$$

$$\boldsymbol{w}^*(\theta, \lambda) \triangleq \begin{cases} \frac{\exp(\theta/\lambda)}{\sum_{j=1}^{d}\exp(\theta_j/\lambda)} & \textit{if } \lambda > 0 \\ \frac{\mathbb{I}[\theta = \max_j \theta_j]}{\sum_{k\in[d]}\mathbb{I}[\theta_k = \max_j \theta_j]} & \textit{if } \lambda = 0 \end{cases} \in \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} \lambda\psi(\boldsymbol{w}) - \langle \boldsymbol{w}, \theta\rangle \subseteq \partial(\lambda\psi)^*_{\mathbb{W}}(\theta). \tag{C.157}$$

Our next corollary concerning optimal ODAFTRL objectives follows directly from Lem. C.13.1.

Corollary C.13.1 (Optimal ODAFTRL objectives)**.** *Instantiate the notation of Lem. C.13.1, and define the functions* $F_t(\boldsymbol{w}, \lambda) \triangleq \lambda\psi(\boldsymbol{w}) + \langle \boldsymbol{g}_{1:t-1}, \boldsymbol{w}\rangle$ *for* $\boldsymbol{w} \in \mathbb{W}$*. Then*

$$-(\lambda\psi)^*_{\mathbb{W}}(-(\boldsymbol{g}_{1:t-1} + \boldsymbol{h})) = \inf_{\boldsymbol{w}\in\mathbb{W}} F_t(\boldsymbol{w}, \lambda) + \langle \boldsymbol{h}, \boldsymbol{w}\rangle \quad \textit{and} \tag{C.158}$$

$$\boldsymbol{w}^*(-(\boldsymbol{g}_{1:t-1} + \boldsymbol{h}), \lambda) = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} F_t(\boldsymbol{w}, \lambda) + \langle \boldsymbol{h}, \boldsymbol{w}\rangle. \tag{C.159}$$

Using Lem. C.13.1 and Cor. C.13.1, we can derive an expression, proved in App. C.13.5, for the AdaHedgeD δ_t updates.

Proposition C.13.1 (AdaHedgeD δ_t)**.** *Instantiate the notation of Thm. 5.7.3, and define the auxiliary hint vector*

$$\hat{\boldsymbol{h}}_t \triangleq \boldsymbol{g}_{t-D:t} + \sigma_t(\boldsymbol{h}_t - \boldsymbol{g}_{t-D:t}) \quad \textit{for} \quad \sigma_t \triangleq \min\Big(\frac{\|\boldsymbol{g}_t\|_*}{\|\boldsymbol{h}_t - \boldsymbol{g}_{t-D:t}\|_*}, 1\Big) \tag{C.160}$$

along with the scalars

$$c_* = \max_{j:\boldsymbol{w}_{t,j}\neq 0} \boldsymbol{h}_{t,j} - \boldsymbol{g}_{t-D:t,j} \quad \textit{and} \quad \hat{c}_* = \max_{j:\hat{\boldsymbol{w}}_{t,j}\neq 0} \hat{\boldsymbol{h}}_{t,j} - \boldsymbol{g}_{t-D:t,j} \tag{C.161}$$

for

$$\bar{\boldsymbol{w}}_t = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t) = \frac{\exp(-\boldsymbol{g}_{1:t}/\lambda_t)}{\sum_{j=1}^{d} \exp(-\boldsymbol{g}_{1:t,j}/\lambda_t)} \quad \textit{and} \tag{C.162}$$

$$\hat{\boldsymbol{w}}_t = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t) + \langle \hat{\boldsymbol{h}}_t - \boldsymbol{g}_{t-D:t}, \boldsymbol{w}\rangle = \frac{\exp(-(\boldsymbol{g}_{1:t-D-1} + \hat{\boldsymbol{h}}_t)/\lambda_t)}{\sum_{j=1}^{d} \exp(-(\boldsymbol{g}_{1:t-D-1,j} + \hat{\boldsymbol{h}}_{t,j})/\lambda_t)} \tag{C.163}$$

by Cor. C.13.1. If $\lambda_t > 0$,

$$\delta_t = \min(\delta_t^{(1)}, \delta_t^{(2)}, \delta_t^{(3)})_+ \quad \textit{for} \tag{C.164}$$

$$\delta_t^{(1)} = F_{t+1}(\boldsymbol{w}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) \tag{C.165}$$

$$= \lambda_t \ln(\sum_{j\in[d]} \boldsymbol{w}_{t,j} \exp((\boldsymbol{h}_{t,j} - \boldsymbol{g}_{t-D:t,j})/\lambda_t)) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t\rangle \tag{C.166}$$

$$= \lambda_t \ln(\sum_{j\in[d]} \boldsymbol{w}_{t,j} \exp((\boldsymbol{h}_{t,j} - \boldsymbol{g}_{t-D:t,j} - c_*)/\lambda_t)) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t\rangle + c_*, \tag{C.167}$$

$$\delta_t^{(2)} = \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \bar{\boldsymbol{w}}_t\rangle, \quad \textit{and} \tag{C.168}$$

$$\delta_t^{(3)} = F_{t+1}(\hat{\boldsymbol{w}}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t\rangle \tag{C.169}$$

$$= \lambda_t \ln(\sum_{j\in[d]} \hat{\boldsymbol{w}}_{t,j} \exp((\hat{\boldsymbol{h}}_{t,j} - \boldsymbol{g}_{t-D:t,j})/\lambda_t)) + \langle \boldsymbol{g}_{t-D:t} - \hat{\boldsymbol{h}}_t, \hat{\boldsymbol{w}}_t\rangle + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t\rangle \tag{C.170}$$

$$= \lambda_t \ln(\sum_{j\in[d]} \hat{\boldsymbol{w}}_{t,j} \exp((\hat{\boldsymbol{h}}_{t,j} - \boldsymbol{g}_{t-D:t,j} - \hat{c}_*)/\lambda_t)) + \langle \boldsymbol{g}_{t-D:t} - \hat{\boldsymbol{h}}_t, \hat{\boldsymbol{w}}_t\rangle + \hat{c}_* + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t\rangle. \tag{C.171}$$

If $\lambda_t = 0$,

$$\delta_t = \min(\delta_t^{(1)}, \delta_t^{(2)}, \delta_t^{(3)})_+ \quad \textit{for} \tag{C.172}$$

$$\delta_t^{(1)} = \langle \boldsymbol{g}_{1:t}, \boldsymbol{w}_t\rangle - \min_{j\in[d]} \boldsymbol{g}_{1:t,j}, \tag{C.173}$$

$$\delta_t^{(2)} = \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \bar{\boldsymbol{w}}_t\rangle, \quad \textit{and} \tag{C.174}$$

$$\delta_t^{(3)} = \langle \boldsymbol{g}_{1:t}, \hat{\boldsymbol{w}}_t\rangle - \min_{j\in[d]} \boldsymbol{g}_{1:t,j} + \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \hat{\boldsymbol{w}}_t\rangle. \tag{C.175}$$

Leveraging these results, we present the pseudocode for the AdaHedgeD and DUB instantiations of ODAFTRL in Algorithm 3.

C.13.2 DORM and DORM+

The DORM and DORM+ algorithms presented in the experiments are implementations of ODAFTRL and DOOMD respectively that play iterates in $\mathbb{W} \triangleq \triangle_{d-1}$ using the default value $\lambda = 1$. Both algorithms use a p-norm regularizer $\psi = \frac{1}{2}\|\cdot\|_p^2$, which is 1-strongly convex with respect to $\|\cdot\| = \sqrt{p-1}\|\cdot\|_p$ [see 152, Lemma 17] with $\|\cdot\|_* = \frac{1}{\sqrt{p-1}}\|\cdot\|_q$. For the thesis experiments, we choose the optimal value $q = \inf_{q' \geq 2} d^{2/q'}(q'-1)$ to obtain $\ln(d)$ scaling in the algorithm regret; for $d = 6$, $p = q = 2$. The update equations for each algorithm are given in the main text by DORM and DORM+ respectively. The optimistic hinters provide delayed gradient hints $\tilde{\boldsymbol{g}}_t$, which are then used to compute regret gradient hints $\tilde{\boldsymbol{r}}_t$, where $\tilde{\boldsymbol{r}}_t = \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_t \rangle - \tilde{\boldsymbol{g}}_t$ and $\boldsymbol{h}_t = \sum_{s=t-D}^{t-1} \tilde{\boldsymbol{r}}_s + \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{w}_{t-1} \rangle - \tilde{\boldsymbol{g}}_t$.

C.13.3 Adaptive Hinting

For the adaptive hinting experiments, we use the DORM+ as both the base and hint learner. For the hint learner with DORM base algorithm, the hint loss function is given by (C.144) with $q = 2$. The plays of the online hinter ω_t are used to generate the hints $\boldsymbol{h}_t$ for the base algorithm using the hint matrix $H_t \in \mathbb{R}^{d \times m}$. The j-th column of H_t contains hinter j's predictions for the cumulative missing regret subgradients $\boldsymbol{r}_{t-D:t}$. The final hint for the base learner is $\boldsymbol{h}_t = H_t \omega_t$. Pseudocode for the adaptive hinter is given in Algorithm 4.

C.13.4 Proof of Lem. C.13.1: Negative entropy properties

The expression of the Fenchel conjugate for $\lambda > 0$ is derived by solving an appropriate constrained convex optimization problem for $\boldsymbol{w} = \triangle_{d-1}$, as shown in [129, Section 6.6]. The value of $\boldsymbol{w}^*(\theta, \lambda) \in \partial(\lambda\psi)^*_{\mathbb{W}}(\theta)$ uses the properties of the Fenchel conjugate [129, 150, Theorem 5.5] and is shown in [129, Theorem 6.6].

C.13.5 Proof of Prop. C.13.1: AdaHedgeD δ_t

First suppose $\lambda_t > 0$. The first term in the min of AdaHedgeD's δ_t setting is derived as follows:

$$\delta_t^{(1)} \triangleq F_{t+1}(\boldsymbol{w}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) \quad \text{by definition (5.25)} \tag{C.176}$$

$$= F_{t-D}(\boldsymbol{w}_t, \lambda_t) + \langle \boldsymbol{h}_t, \boldsymbol{w}_t \rangle + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle - \inf_{\boldsymbol{w} \in \mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t) \quad \text{by definition of } \bar{\boldsymbol{w}}_t \tag{C.177}$$

$$= F_{t-D}(\boldsymbol{w}_t, \lambda_t) + \langle \boldsymbol{h}_t, \boldsymbol{w}_t \rangle + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle + \lambda_t \psi_{\mathbb{W}}^*(-\boldsymbol{g}_{1:t}/\lambda_t) \quad \text{by Cor. C.13.1} \tag{C.178}$$

$$= \lambda_t \psi_{\mathbb{W}}^*(-\boldsymbol{g}_{1:t}/\lambda_t) - \lambda_t \psi_{\mathbb{W}}^*((-\boldsymbol{h}_t - \boldsymbol{g}_{1:t-D-1})/\lambda_t) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.179}$$

$$\text{because } \boldsymbol{w}_t \in \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} F_{t-D}(\boldsymbol{w}_t, \lambda_t) + \langle \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.180}$$

$$= \lambda_t (\ln(\sum_{j=1}^{d} \exp(-\boldsymbol{g}_{1:t,j}/\lambda_t)) - \lambda_t (\ln(\sum_{j=1}^{d} \exp((-\boldsymbol{g}_{1:t-D-1,j} - \boldsymbol{h}_{t,j})/\lambda_t)) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.181}$$

by Lem. C.13.1

$$= \lambda_t \ln \left(\sum_{j=1}^{d} \frac{\exp(-\boldsymbol{g}_{1:t,j}/\lambda_t)}{\sum_{j=1}^{d} \exp((-\boldsymbol{g}_{1:t-D-1,j} - \boldsymbol{h}_{t,j})/\lambda_t)} \right) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.182}$$

$$= \lambda_t \ln \left(\sum_{j=1}^{d} \frac{\exp((-\boldsymbol{g}_{1:t-D-1,j} - \boldsymbol{h}_{t,j})/\lambda_t) \exp((\boldsymbol{h}_{t,j} - \boldsymbol{g}_{t-D:t,j})/\lambda_t)}{\sum_{j=1}^{d} \exp((-\boldsymbol{g}_{1:t-D-1,j} - \boldsymbol{h}_{t,j})/\lambda_t)} \right) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.183}$$

$$= \lambda_t \ln \left(\sum_{j=1}^{d} \boldsymbol{w}_{t,j} \exp((\boldsymbol{h}_{t,j} - \boldsymbol{g}_{t-D:t,j})/\lambda_t) \right) + \langle \boldsymbol{g}_{t-D:t} - \boldsymbol{h}_t, \boldsymbol{w}_t \rangle \tag{C.184}$$

by the expression for $\boldsymbol{w}_t$ in Cor. C.13.1.

The expression for the third term in the min of AdaHedgeD's δ_t setting follows from identical reasoning.

Now suppose $\lambda_t = 0$. We have

$$\delta_t^{(1)} \triangleq F_{t+1}(\boldsymbol{w}_t, \lambda_t) - F_{t+1}(\bar{\boldsymbol{w}}_t, \lambda_t) \quad \text{by definition (5.25)} \tag{C.185}$$

$$= \langle \boldsymbol{g}_{1:t}, \boldsymbol{w}_t \rangle - \inf_{\boldsymbol{w} \in \mathbb{W}} F_{t+1}(\boldsymbol{w}, \lambda_t) \quad \text{by definition of } \bar{\boldsymbol{w}}_t \tag{C.186}$$

$$= \langle \boldsymbol{g}_{1:t}, \boldsymbol{w}_t \rangle - \min_{j \in [d]} \boldsymbol{g}_{1:t,j} \quad \text{by Cor. C.13.1.} \tag{C.187}$$

Identical reasoning yields the advertised expression for the third term.

C.14 Extension to Variable and Unbounded Delays

In this section we detail how our main results generalize to the case of variable and potentially unbounded delays. For each time t, we define last(t) as the largest index s for which $\boldsymbol{g}_{1:s}$ is observable at time t (that is, available for constructing $\boldsymbol{w}_t$) and first(t) as the first time s at which $\boldsymbol{g}_{1:t}$ is observable at time s (that is, available for constructing $\boldsymbol{w}_s$).

C.14.1 Regret of DOOMD with variable delays

Consider the DOOMD variable-delay generalization

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathbb{W}} \langle \boldsymbol{g}_{\text{last}(t)+1:\text{last}(t+1)} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t, \boldsymbol{w} \rangle + \mathcal{B}_{\lambda\psi}(\boldsymbol{w}, \boldsymbol{w}_t) \tag{DOOMD with variable delays}$$

$$\text{with} \quad \boldsymbol{h}_0 \triangleq \boldsymbol{0} \quad \text{and arbitrary} \quad \boldsymbol{w}_0.$$

We first note that DOOMD with variable delays is an instance of SOOMD respectively with a "bad" choice of optimistic hint $\tilde{\boldsymbol{g}}_{t+1}$ that deletes the unobserved loss subgradients $\boldsymbol{g}_{\text{last}(t+1)+1:t}$.

Lemma C.14.1 (DOOMD with variable delays is SOOMD with a bad hint)**.** DOOMD with variable delays *is* SOOMD *with* $\tilde{\boldsymbol{g}}_{t+1} = \tilde{\boldsymbol{g}}_t + \boldsymbol{g}_{\text{last}(t)+1:\text{last}(t+1)} - \boldsymbol{g}_t + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t = \boldsymbol{h}_{t+1} + \sum_{s=1}^{t} \boldsymbol{g}_{\text{last}(s)+1:\text{last}(s+1)} - \boldsymbol{g}_s. = \boldsymbol{h}_{t+1} - \boldsymbol{g}_{\text{last}(t+1)+1:t}.$

The following result now follows immediately from Thm. 5.4.2 and Lem. C.14.1.

Theorem C.14.1 (Regret of DOOMD with variable delays)**.** *If ψ is differentiable and $\boldsymbol{h}_{T+1} \triangleq \boldsymbol{g}_{\text{last}(T+1)+1:T}$, then, for all $\boldsymbol{u} \in \mathbb{W}$, the* DOOMD with variable delays *iterates $\boldsymbol{w}_t$*

satisfy

$$\text{Regret}_T(\boldsymbol{u}) \leq \mathcal{B}_{\lambda\psi}(\boldsymbol{u}, \boldsymbol{w}_0) + \frac{1}{\lambda}\sum_{t=1}^{T} \boldsymbol{b}_{t,O}^2, \quad \textit{for} \tag{C.188}$$

$$\boldsymbol{b}_{t,O}^2 \triangleq \text{huber}(\|\boldsymbol{h}_t - \sum_{s=\text{last}(t)+1}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_{\text{last}(t)+1:\text{last}(t+1)} + \boldsymbol{h}_{t+1} - \boldsymbol{h}_t\|_*). \tag{C.189}$$

C.14.2 Regret of ODAFTRL with variable delays

Consider the ODAFTRL variable-delay generalization

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{W}} \langle \boldsymbol{g}_{1:\text{last}(t+1)} + \boldsymbol{h}_{t+1}, \boldsymbol{w}\rangle + \lambda_{t+1}\psi(\boldsymbol{w}). \quad \text{(ODAFTRL with variable delays)}$$

Since ODAFTRL with variable delays is an instance of OAFTRL with $\tilde{\boldsymbol{g}}_{t+1} = \boldsymbol{h}_{t+1} - \sum_{s=\text{last}(t+1)+1}^{t} \boldsymbol{g}_s$, the following result follows immediately from the OAFTRL regret bound, Thm. C.2.1.

Theorem C.14.2 (Regret of ODAFTRL with variable delays)**.** *If ψ is nonnegative and λ_t is non-decreasing in t, then, $\forall \boldsymbol{u} \in \mathbb{W}$, the* ODAFTRL with variable delays *iterates $\boldsymbol{w}_t$ satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda_T\psi(\boldsymbol{u}) + \sum_{t=1}^{T} \min(\frac{\boldsymbol{b}_{t,F}}{\lambda_t}, \boldsymbol{a}_{t,F}) \quad \textit{with} \tag{C.190}$$

$$\boldsymbol{b}_{t,F} \triangleq \text{huber}(\|\boldsymbol{h}_t - \sum_{s=\text{last}(t)+1}^{t} \boldsymbol{g}_s\|_*, \|\boldsymbol{g}_t\|_*) \quad \textit{and} \tag{C.191}$$

$$\boldsymbol{a}_{t,F} \triangleq \text{diam}(\mathbb{W}) \min\left(\|\boldsymbol{h}_t - \sum_{s=\text{last}(t)+1}^{t} \boldsymbol{g}_s\|, \|\boldsymbol{g}_t\|_*\right). \tag{C.192}$$

C.14.3 Regret of DUB with variable delays

Consider the DUB variable-delay generalization

$$\alpha\lambda_{t+1} = 2 \max_{j\leq \text{last}(t+1)-1} \boldsymbol{a}_{\text{last}(j+1)+1:j,F} + \sqrt{\textstyle\sum_{i=1}^{\text{last}(t+1)} \boldsymbol{a}_{i,F}^2 + 2\alpha \boldsymbol{b}_{i,F}}.$$

(DUB with variable delays)

Theorem C.14.3 (Regret of DUB with variable delays)**.** *Fix $\alpha > 0$, and, for $\boldsymbol{a}_{t,F}, \boldsymbol{b}_{t,F}$ as in* (C.191)*, consider the* DUB with variable delays *sequence. If ψ is nonnegative, then, for* *all* $\boldsymbol{u} \in \mathbb{W}$*, the* ODAFTRL with variable delays *iterates* $\boldsymbol{w}_t$ *satisfy*

$$\text{Regret}_T(\boldsymbol{u}) \leq \big(\frac{\psi(\boldsymbol{u})}{\alpha} + 1\big) \tag{C.193}$$

$$\big(2 \max_{t\in[T]} \boldsymbol{a}_{\text{last}(t)+1:t-1,F} + \sqrt{\textstyle\sum_{t=1}^T \boldsymbol{a}_{t,F}^2 + 2\alpha \boldsymbol{b}_{t,F}}\big) \tag{C.194}$$

Proof. Fix any $\boldsymbol{u} \in \mathbb{W}$. By Thm. C.14.2, ODAFTRL with variable delays admits the regret bound

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda_T \psi(\boldsymbol{u}) + \sum_{t=1}^T \min(\frac{1}{\lambda_t} \boldsymbol{b}_{t,F}, \boldsymbol{a}_{t,F}). \tag{C.195}$$

To control the second term in this bound, we apply the following lemma proved in App. C.8.1.

Lemma C.14.2 (DUB with variable delays-style tuning bound)**.** *Fix any* $\alpha > 0$ *and any non-negative sequences* $(a_t)_{t=1}^T$*,* $(b_t)_{t=1}^T$*. If* $(\lambda_t)_{t\geq1}$ *is non-decreasing and*

$$\Delta_{t+1}^* \triangleq 2 \max_{j\leq \text{last}(t+1)-1} a_{\text{last}(j+1)+1:j} + \sqrt{\sum_{i=1}^{\text{last}(t+1)} a_i^2 + 2\alpha b_i} \leq \alpha\lambda_{t+1} \quad \textit{for each} \quad t \tag{C.196}$$

then

$$\sum_{t=1}^T \min(b_t/\lambda_t, a_t) \leq \Delta_{\text{first}(T)}^* \leq \alpha\lambda_{\text{first}(T)}. \tag{C.197}$$

■

Since $T \leq \text{first}(T)$, $\lambda_T \leq \lambda_{\text{first}(T)}$, and $\text{last}(\text{first}(T)) = T$, the result now follows by setting $a_t = \boldsymbol{a}_{t,F}$ and $b_t = \boldsymbol{b}_{t,F}$, so that

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda_T \psi(\boldsymbol{u}) + \alpha\lambda_{\text{first}(T)} \leq (\psi(\boldsymbol{u}) + \alpha)\lambda_{\text{first}(T)}. \tag{C.198}$$

C.14.4 Proof of Lem. C.14.2: DUB with variable delays-style tuning bound

We prove the claim

$$\Delta_t \triangleq \sum_{i=1}^t \min(b_i/\lambda_i, a_i) \leq \Delta_{\text{first}(t)}^* \leq \alpha\lambda_{\text{first}(t)} \tag{C.199}$$

by induction on t.

Base case For $t = 1$, since $\mathrm{last}(\mathrm{first}(t)) \ge t$, we have

$$\sum_{i=1}^{t} \min(b_i/\lambda_i, a_i) \le a_1 \le 2 \max_{j \le t-1} a_{\mathrm{last}(j+1)+1:j} + \sqrt{\sum_{i=1}^{t} a_i^2 + 2\alpha b_i} \tag{C.200}$$

$$\le 2 \max_{j \le \mathrm{last}(\mathrm{first}(t))-1} a_{\mathrm{last}(j+1)+1:j} + \sqrt{\sum_{i=1}^{\mathrm{last}(\mathrm{first}(t))} a_i^2 + 2\alpha b_i} = \Delta^*_{\mathrm{first}(t)} \le \alpha\lambda_{\mathrm{first}(t)} \tag{C.201}$$

confirming the base case.

Inductive step Now fix any $t + 1 \ge 2$ and suppose that

$$\Delta_i \le \Delta^*_{\mathrm{first}(i)} \le \alpha\lambda_{\mathrm{first}(i)} \tag{C.202}$$

for all $1 \le i \le t$. Since $\mathrm{first}(\mathrm{last}(i+1)) \le i+1$ and λ_s is non-decreasing in s, we apply this inductive hypothesis to deduce that, for each $0 \le i \le t$,

$$\Delta_{i+1}^2 - \Delta_i^2 = (\Delta_i + \min(b_{i+1}/\lambda_{i+1}, a_{i+1}))^2 - \Delta_i^2 = 2\Delta_i \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 \tag{C.203}$$

$$= 2\Delta_{\mathrm{last}(i+1)} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2(\Delta_i - \Delta_{\mathrm{last}(i+1)}) \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 \tag{C.204}$$

$$= 2\Delta_{\mathrm{last}(i+1)} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2 \sum_{j=\mathrm{last}(i+1)+1}^{i} \min(b_j/\lambda_j, a_j) \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + \min(b_{i+1}/\lambda_{i+1}, a_{i+1})^2 \tag{C.205}$$

$$\le 2\alpha\lambda_{\mathrm{first}(\mathrm{last}(i+1))} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2a_{\mathrm{last}(i+1)+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + a_{i+1}^2 \tag{C.206}$$

$$\le 2\alpha\lambda_{i+1} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + 2a_{\mathrm{last}(i+1)+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) + a_{i+1}^2 \tag{C.207}$$

$$\le 2\alpha b_{i+1} + a_{i+1}^2 + 2a_{\mathrm{last}(i+1)+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}). \tag{C.208}$$

Now, we sum this inequality over $i = 0, \dots, t$, to obtain

$$\Delta_{t+1}^2 \le \sum_{i=0}^{t}(2\alpha b_{i+1} + a_{i+1}^2) + 2\sum_{i=0}^{t} a_{\text{last}(i+1)+1:i} \min(b_{i+1}/\lambda_{i+1}, a_{i+1}) \tag{C.209}$$

$$= \sum_{i=1}^{t+1}(2\alpha b_i + a_i^2) + 2\sum_{i=1}^{t+1} a_{\text{last}(i+1):i-1} \min(b_i/\lambda_i, a_i) \tag{C.210}$$

$$\le \sum_{i=1}^{t+1}(a_i^2 + 2\alpha b_i) + 2\max_{j\le t} a_{\text{last}(j+1)+1:j} \sum_{i=1}^{t+1} \min(b_i/\lambda_i, a_i) \tag{C.211}$$

$$= \sum_{i=1}^{t+1}(a_i^2 + 2\alpha b_i) + 2\Delta_{t+1} \max_{j\le t} a_{\text{last}(j+1)+1:j}. \tag{C.212}$$

We now solve this quadratic inequality, apply the triangle inequality, and invoke the relation $\text{last}(\text{first}(t+1)) \ge t+1$ to conclude that

$$\Delta_{t+1} \le \max_{j\le t} a_{\text{last}(j+1)+1:j} + \frac{1}{2}\sqrt{(2\max_{j\le t} a_{\text{last}(j+1)+1:j})^2 + 4\sum_{i=1}^{t+1} a_i^2 + 2\alpha b_i} \tag{C.213}$$

$$\le 2\max_{j\le t} a_{\text{last}(j+1)+1:j} + \sqrt{\sum_{i=1}^{t+1} a_i^2 + 2\alpha b_i} \tag{C.214}$$

$$\le 2 \max_{j\le \text{last}(\text{first}(t+1))-1} a_{\text{last}(j+1)+1:j} + \sqrt{\sum_{i=1}^{\text{last}(\text{first}(t+1))} a_i^2 + 2\alpha b_i} = \Delta^*_{\text{first}(t+1)} \le \alpha\lambda_{\text{first}(t+1)}. \tag{C.215}$$

C.14.5 Regret of AdaHedgeD with variable delays

Consider the AdaHedgeD variable-delay generalization

$$\lambda_{t+1} = \frac{1}{\alpha}\sum_{s=1}^{\text{last}(t+1)} \delta_s \quad \text{for} \quad \delta_t \quad \text{defined in (5.25).} \qquad \text{(AdaHedgeD with variable delays)}$$

Theorem C.14.4 (Regret of AdaHedgeD with variable delays)**.** *Fix $\alpha > 0$, and consider the* AdaHedgeD with variable delays *sequence. If ψ is nonnegative, then, for all $\boldsymbol{u} \in \mathbb{W}$, the*

ODAFTRL with variable delays *iterates satisfy*

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \big(\frac{\psi(\boldsymbol{u})}{\alpha} + 1\big) \tag{C.216}$$

$$\big(2 \max_{t \in [T]} \boldsymbol{a}_{\mathrm{last}(t+1)+1:t,F} + \sqrt{\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha \boldsymbol{b}_{t,F}}\big). \tag{C.217}$$

Proof. Fix any $\boldsymbol{u} \in \mathbb{W}$, and for each t, define $\lambda'_{t+1} = \frac{1}{\alpha}\sum_{s=1}^{t} \delta_s$ so that $\alpha(\lambda'_{t+1} - \lambda'_t) = \delta_t$. Since the AdaHedgeD with variable delays regularization sequence $(\lambda_t)_{t\geq 1}$ is non-decreasing, $\mathrm{last}(T) \leq T$, and hence $\lambda_T \leq \lambda'_{T+1}$, Thm. C.2.1 gives the regret bound

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda_T \psi(\boldsymbol{u}) + \sum_{t=1}^{T} \delta_t \leq \lambda_T \psi(\boldsymbol{u}) + \alpha \lambda'_{T+1} \leq (\psi(\boldsymbol{u}) + \alpha)\lambda'_{T+1} \tag{C.218}$$

and the proof of Thm. C.2.1 gives the upper estimate (C.32):

$$\delta_t \leq \min\left(\frac{\boldsymbol{b}_{t,F}}{\lambda_t}, \boldsymbol{a}_{t,F}\right) \quad \text{for all} \quad t \in [T]. \tag{C.219}$$

Hence, it remains to bound λ'_{T+1}. We have

$$\begin{align}
\alpha {\lambda'_{T+1}}^2 &= \sum_{t=1}^{T} \alpha({\lambda'_{t+1}}^2 - {\lambda'_t}^2) = \sum_{t=1}^{T} \big(\alpha(\lambda'_{t+1} - \lambda'_t)^2 + 2\alpha(\lambda'_{t+1} - \lambda'_t)\lambda'_t\big) \tag{C.220}\\
&= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda'_t\big) \quad \text{by the definition of } \lambda'_{t+1} \tag{C.221}\\
&= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_t + 2\delta_t(\lambda'_t - \lambda_t)\big) \tag{C.222}\\
&\le \sum_{t=1}^{T} \left(\delta_t^2/\alpha + 2\delta_t\lambda_t + 2\delta_t \max_{t\in[T]}(\lambda'_t - \lambda_t)\right) \tag{C.223}\\
&= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_t\big) + 2\alpha\lambda'_{T+1} \max_{t\in[T]}(\lambda'_t - \lambda_t) \tag{C.224}\\
&= \sum_{t=1}^{T} \big(\delta_t^2/\alpha + 2\delta_t\lambda_t\big) + 2\lambda'_{T+1} \max_{t\in[T]} \delta_{\mathrm{last}(t+1)+1:t} \tag{C.225}\\
&\le \sum_{t=1}^{T} \big(\boldsymbol{a}_{t,F}^2/\alpha + 2\boldsymbol{b}_{t,F}\big) + 2\lambda'_{T+1} \max_{t\in[T]} \boldsymbol{a}_{\mathrm{last}(t+1)+1:t,F} \quad \text{by (C.219).} \tag{C.226}
\end{align}$$

Solving the above quadratic inequality for λ'_{T+1} and applying the triangle inequality, we find

$$\begin{align}
\alpha\lambda'_{T+1} &\le \max_{t\in[T]} \boldsymbol{a}_{\mathrm{last}(t+1)+1:t,F} + \frac{1}{2}\sqrt{4(\max_{t\in[T]} \boldsymbol{a}_{\mathrm{last}(t+1)+1:t,F})^2 + 4\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha\boldsymbol{b}_{t,F}} \tag{C.227}\\
&\le 2\max_{t\in[T]} \boldsymbol{a}_{\mathrm{last}(t+1)+1:t,F} + \sqrt{\sum_{t=1}^{T} \boldsymbol{a}_{t,F}^2 + 2\alpha\boldsymbol{b}_{t,F}}. \tag{C.228}
\end{align}$$

■

```
Parameter $\alpha = \sup_{\boldsymbol{u} \in \triangle_{d-1}} \psi(\boldsymbol{u}) = \ln(d)$
Initial regularization weight: $\lambda_0 = 0$
**if** `tuning` is DUB **then**
  Initial regularization sum: $\Delta_0 = 0$
  Initial maximum: $\boldsymbol{a}^{\max} = 0$
**end if**
Initial subgradient sum: $\boldsymbol{g}_{1:1} = \boldsymbol{0} \in \mathbb{R}^d$
Dummy losses and iterates: $\boldsymbol{g}_{-D} = \cdots = \boldsymbol{g}_0 = \boldsymbol{0} \in \mathbb{R}^d$, $\boldsymbol{w}_{-D} = \cdots = \boldsymbol{w}_0 = \boldsymbol{0} \in \mathbb{R}^d$
**for** $t = 1, \ldots, T$ **do**
  Receive hint $\boldsymbol{h}_t \in \mathbb{R}^d$
  Output $\boldsymbol{w}_t = \arg\min_{\boldsymbol{w} \in \mathbb{W}} F_{t-D}(\boldsymbol{w}, \lambda_t) + \langle \boldsymbol{h}_t, \boldsymbol{w} \rangle$ as in Cor. C.13.1
  Receive $\boldsymbol{g}_{t-D} \in \mathbb{R}^d$ and pay $\langle \boldsymbol{g}_{t-D}, \boldsymbol{w}_{t-D} \rangle$
  Update subgradient sum $\boldsymbol{g}_{1:t-D} = \boldsymbol{g}_{1:t-D-1} + \boldsymbol{g}_{t-D}$
  **if** `tuning` is AdaHedgeD **then**
    Compute the auxiliary play $\bar{\boldsymbol{w}}_{t-D} = \arg\min_{\boldsymbol{w} \in \mathbb{W}} F_{t-D+1}(\boldsymbol{w}, \lambda_{t-D})$ as in Cor. C.13.1
    Compute the auxiliary regret term $\delta^{(1)}_{t-D} = F_{t-D+1}(\boldsymbol{w}_{t-D}, \lambda_{t-D}) - F_{t-D+1}(\bar{\boldsymbol{w}}_{t-D}, \lambda_{t-D})$ as in Prop. C.13.1
    Compute the drift term $\delta^{(2)}_{t-D} = \langle \boldsymbol{g}_{t-D}, \boldsymbol{w}_{t-D} - \bar{\boldsymbol{w}}_{t-D} \rangle$
    Compute the auxiliary hint (C.160) $\hat{\boldsymbol{h}}_{t-D} \triangleq \boldsymbol{g}_{t-2D:t-D} + \min(\frac{\|\boldsymbol{g}_{t-D}\|_*}{\|\boldsymbol{h}_{t-D} - \boldsymbol{g}_{t-2D:t-D}\|_*}, 1)(\boldsymbol{h}_{t-D} - \boldsymbol{g}_{t-2D:t-D})$
    Compute the auxiliary play $\hat{\boldsymbol{w}}_{t-D} = \arg\min_{\boldsymbol{w} \in \mathbb{W}} F_{t-D+1}(\boldsymbol{w}, \lambda_{t-D}) + \langle \hat{\boldsymbol{h}}_{t-D} - \boldsymbol{g}_{t-2D:t-D}, \boldsymbol{w} \rangle$ as in Cor. C.13.1
    Compute the regret term $\delta^{(3)}_{t-D} = F_{t-D+1}(\hat{\boldsymbol{w}}_{t-D}, \lambda_{t-D}) - F_{t-D+1}(\bar{\boldsymbol{w}}_{t-D}, \lambda_{t-D}) + \langle \boldsymbol{g}_{t-D}, \boldsymbol{w}_{t-D} - \hat{\boldsymbol{w}}_{t-D} \rangle$ as in Prop. C.13.1
    Update $\lambda_{t+1} = \lambda_t + \frac{1}{\alpha} \min(\delta^{(1)}_{t-D}, \delta^{(2)}_{t-D}, \delta^{(3)}_{t-D})_+$ as in (5.25)
  **else if** `tuning` is DUB **then**
    Compute $\boldsymbol{a}_{t-D,F} = 2\min\left(\|\boldsymbol{g}_{t-D}\|_\infty, \|\boldsymbol{h}_{t-D} - \sum_{s=t-2D}^{t-D} \boldsymbol{g}_s\|_\infty\right)$ as in (5.18)
    Compute $\boldsymbol{b}_{t-D,F} = \frac{1}{2}\|\boldsymbol{h}_{t-D} - \sum_{s=t-2D}^{t-D} \boldsymbol{g}_s\|_\infty^2 - \frac{1}{2}(\|\boldsymbol{h}_{t-D} - \sum_{s=t-2D}^{t-D} \boldsymbol{g}_s\|_\infty - \|\boldsymbol{g}_{t-D}\|_\infty)_+^2$ as in (5.18)
    Update $\Delta_{t+1} = \Delta_t + \boldsymbol{a}^2_{t-D,F} + 2\alpha \boldsymbol{b}_{t-D,F}$
    Update maximum $\boldsymbol{a}^{\max} = \max(\boldsymbol{a}^{\max}, \boldsymbol{a}_{t-2D:t-D-1,F})$
    Update $\lambda_{t+1} = \frac{1}{\alpha}(2\boldsymbol{a}^{\max} + \sqrt{\Delta_{t+1}})$ as in DUB
  **end if**
**end for**
```

Algorithm 3: ODAFTRL *with* $\mathbb{W} = \triangle_{d-1}$*,* $\psi(\boldsymbol{w}) = \sum_{j=1}^{d} \boldsymbol{w}_j \ln \boldsymbol{w}_j + \ln(d)$*, delay* $D \geq 0$*, and tuning strategy* `tuning` $=$ DUB *or* AdaHedgeD

```
Subgradient vector: $\boldsymbol{g}_{-D}, \cdots \boldsymbol{g}_0 = \mathbf{0} \in \mathbb{R}^d$
Meta-subgradient vector: $\gamma_{-D}, \cdots \gamma_0 = \mathbf{0} \in \mathbb{R}^m$
Initial instantaneous regret: $\boldsymbol{r}_{-D} = \mathbf{0} \in \mathbb{R}^d$
Initial instantaneous meta-regret: $\rho_{-D} = \mathbf{0} \in \mathbb{R}^m$
Initial hint $\boldsymbol{h}_0 = \mathbf{0} \in \mathbb{R}^d$
Initial orthant meta-vector: $\tilde{\omega}_0 = \mathbf{0} \in \mathbb{R}^m$
for $t = 1, \ldots, T$ do
    // Update online hinter using DORM+ with $q = 2$
    Find optimal unnormalized hint combination vector $\tilde{\omega}_t = \max(\mathbf{0}, \tilde{\omega}_{t-1} + \rho_{t-D-1})$
    Normalize: $\omega_t = \begin{cases} \mathbf{1}/m & \text{if } \tilde{\omega}_t = \mathbf{0} \\ \tilde{\omega}_t / \langle \mathbf{1}, \tilde{\omega}_t \rangle & \text{otherwise} \end{cases}$
    Receive hint matrix: $H_t \in \mathbb{R}^{d \times m}$ in which each column is a hint for $\sum_{s=t-D}^{t} \boldsymbol{r}_s$
    Output hint $\boldsymbol{h}_t = H_t \omega_t$
    // Update DORM+ base learner and get next play
    Output $\boldsymbol{w}_t = \text{DORM+}(\boldsymbol{g}_{t-D-1}, \boldsymbol{h}_t)$
    Receive $\boldsymbol{g}_{t-D} \in \mathbb{R}^d$ and pay $\langle \boldsymbol{g}_{t-D}, \boldsymbol{w}_{t-D} \rangle$
    Compute instantaneous regret $\boldsymbol{r}_{t-D} = \mathbf{1}\langle \boldsymbol{g}_{t-D}, \boldsymbol{w}_{t-D} \rangle - \boldsymbol{g}_{t-D}$
    Compute hint meta-subgradient $\gamma_{t-D} \in \partial l_{t-D}(\omega_{t-D}) \in \mathbb{R}^m$ as in (C.146)
    Compute instantaneous hint regret $\rho_{t-D} = \mathbf{1}\langle \gamma_{t-D}, \omega_{t-D} \rangle - \gamma_{t-D}$
end for
```

Algorithm 4: Learning to hint with DORM+ *(q=2) hint learner,* DORM+ *base learner, and delay* $D \geq 0$

Bibliography

[1] Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[2] Cameron Allen, Tim Klinger, George Konidaris, Matthew Riemer, and Gerald Tesauro. Finding Macro-Actions with Disentangled Effects for Efficient Planning with the Goal-Count Heuristic. *arXiv preprint*, 2020.

[3] Christopher Amato, George Konidaris, Leslie P Kaelbling, and Jonathan P How. Modeling and planning with macro-actions in decentralized POMDPs. *Journal of Artificial Intelligence Research*, 64:817–859, 2019.

[4] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A POMDP extension with belief-dependent rewards. *Advances in neural information processing systems*, 23, 2010.

[5] Troy Arcomano, Istvan Szunyogh, Jaideep Pathak, Alexander Wikner, Brian R Hunt, and Edward Ott. A Machine Learning-Based Global Atmospheric Forecast Model. *Geophysical Research Letters*, 47(9), 2020.

[6] Akash Arora, P Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. Multi-modal active perception for information gathering in science missions. In *Proc. Int. Symp. Auton. Robots*, pages 1–27, 2017.

[7] Karl J Astrom. Optimal control of Markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.*, 10:174–205, 1965.

[8] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in

a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th annual foundations of computer science*, pages 322–331. IEEE, 1995.

[9] David Auger, Adrien Couetoux, and Olivier Teytaud. Continuous upper confidence trees with polynomial exploration–consistency. In *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, pages 194–209, 2013.

[10] Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3): 211–246, 2001.

[11] Akhil Bagaria and George Konidaris. Option Discovery using Deep Skill Chaining. In *The NeurIPS 2019 Workshop on Deep Reinforcement Learning*, 2019.

[12] Haoyu Bai, David Hsu, Wee Sun Lee, and Vien A Ngo. Monte Carlo value iteration for continuous-state POMDPs. In *Algorithmic foundations of robotics IX*, pages 175–191. Springer, 2010.

[13] Anastazia T Banaszak and Michael P Lesser. Effects of solar ultraviolet radiation on coral reef organisms. *Photochem. Photobiol. Sci.*, 8(9):1276–1294, 2009.

[14] Stace E Beaulieu, Edward T Baker, and Christopher R German. Where are the undiscovered hydrothermal vents on oceanic spreading ridges? *Deep Sea Research Part II: Topical Studies in Oceanography*, 121:202–212, 2015.

[15] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

[16] Andrew A. Bennett and John J. Leonard. Behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles. *IEEE J. Ocean. Eng.*, 25(2):213–226, apr 2000.

[17] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[18] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995.

[19] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2147–2154, 2012.

[20] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[21] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[22] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[23] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[24] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Solving continuous POMDPs: Value iteration with incremental learning of an efficient space representation. In *International Conference on Machine Learning*, pages 370–378. PMLR, 2013.

[25] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games*, 4(1):1–43, Mar. 2012.

[26] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games.* Cambridge university press, 2006.

[27] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[28] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, pages 6.1–6.20, Edinburgh, Scotland, 25–27 Jun 2012.

[29] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Proc. Field Service Robot.*, pages 203–209, 1998.

[30] David A Clague, Jennifer B Paduan, David W Caress, Craig L Moyer, Brian T Glazer, and Dana R Yoerger. Structure of Lō ‘ihi Seamount, Hawai‘i and Lava flow morphology from high-resolution mapping. *Frontiers in Earth Science*, 7:58, 2019.

[31] J. Cohen, D. Coumou, J. Hwang, L. Mackey, P. Orenstein, S. Totz, and E. Tziperman. S2S reboot: An argument for greater inclusion of machine learning in subseasonal to seasonal (S2S) forecasts. *WIREs Climate Change*, 10, 2018.

[32] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration. In *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, pages 225–240, 2013.

[33] John B Corliss, Jack Dymond, Louis I Gordon, John M Edmond, Richard P von Herzen, Robert D Ballard, Kenneth Green, David Williams, Arnold Bainbridge, Kathy Crane, et al. Submarine thermal springs on the Galapagos Rift. *Science*, 203(4385): 1073–1083, 1979.

[34] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous upper confidence trees. In *Proc. Int. Conf. Learn. Intell. Optim.*, pages 433–445, 2011.

[35] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1999.

[36] Ashok Cutkosky. Combining online learning guarantees. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 895–913, Phoenix, USA, 25–28 Jun 2019. PMLR.

[37] Patrick Dallaire, Camille Besse, Stephane Ross, and Brahim Chaib-draa. Bayesian reinforcement learning in continuous POMDPs with Gaussian processes. In *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 2604–2609, 2009.

[38] John M Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 2012.

[39] Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15 (37):1281–1316, 2014.

[40] Tim Erven, Wouter M Koolen, Steven Rooij, and Peter Grünwald. Adaptive hedge. In *Advances in Neural Information Processing Systems*, volume 24, pages 1656–1664, 2011.

[41] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint*, 2018.

[42] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5363–5371, May 2021.

[43] Mathieu Fehr, Olivier Buffet, Vincent Thomas, and Jilles Steeve Dibangoye. ρ-POMDPs have Lipschitz-Continuous ϵ-Optimal Value Functions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6933–6943, 2018.

[44] Johannes Fischer and Ömer Sahin Tas. Information particle filter tree: An online algorithm for POMDPs with belief-based rewards on continuous domains. In *International Conference on Machine Learning*, pages 3177–3187. PMLR, 2020.

[45] Genevieve Flaspohler, Victoria Preston, Anna PM Michel, Yogesh Girdhar, and Nicholas Roy. Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters*, pages 3782–3789, 2019.

[46] Genevieve Flaspohler, Nicholas A Roy, and John W Fisher III. Belief-dependent macro-action discovery in POMDPs using the value of information. *Advances in Neural Information Processing Systems*, 33:11108–11118, 2020.

[47] Genevieve Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Online learning with optimism and delay. In *International Conference on Machine Learning*. PMLR, 2021.

[48] Genevieve Flaspohler, Francesco Orabona, Judah Cohen, Soukayna Mouatadid, Miruna Oprescu, Paulo Orenstein, and Lester Mackey. Replication Data for: Online Learning with Optimism and Delay, 2021.

[49] Neha Priyadarshini Garg, David Hsu, and Wee Sun Lee. DESPOT-Alpha: Online POMDP Planning with Large State and Observation Spaces. In *Robotics: Science and Systems*, 2019.

[50] C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3): 265–299, 2003.

[51] Yogesh Girdhar, Philippe Giguere, and Gregory Dudek. Autonomous adaptive exploration using realtime online spatiotemporal topic modeling. *The International Journal of Robotics Research*, 33(4):645–657, 2014.

[52] Yogesh Girdhar, Brian Claus, Seth McCammon, Stewart Jamieson, John San Soucie, Levi Cai, Nathan McGuire, and Stefano Suman. WARPLab. http://warp.whoi.edu/, 2022.

[53] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in Gaussian processes. In *Proc. 22nd Int. Conf. Mach. Learn.*, pages 265–272. ACM, 2005.

[54] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[55] Noor Titan Putri Hartono, Janak Thapa, Armi Tiihonen, Felipe Oviedo, Clio Batali, Jason J Yoo, Zhe Liu, Ruipeng Li, David Fuertes Marrón, Moungi G Bawendi, et al. How machine learning can help select capping layers to suppress perovskite degradation. *Nature communications*, 11(1):1–9, 2020.

[56] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[57] Kris Hauser. Randomized belief-space replanning in partially observable continuous spaces. In *Algorithmic Foundations of Robotics IX*, pages 193–209. Springer, 2010.

[58] David S Hayden, Jason Pacheco, and John W Fisher. Nonparametric object and parts modeling with lie group dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7426–7435, 2020.

[59] Ruijie He, Emma Brunskill, and Nicholas Roy. Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, 40:523–570, 2011.

[60] Sijie He, Xinyan Li, Timothy DelSole, Pradeep Ravikumar, and Arindam Banerjee. Sub-seasonal climate forecasting via machine learning: Challenges, analysis, and advances. *arXiv preprint arXiv:2006.07972*, 2020.

[61] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, pages 918–926, 2014.

[62] Gregory Hitz, Enric Galceran, Marie-Ève Garneau, François Pomerleau, and Roland Siegwart. Adaptive continuous-space informative path planning for online environmental monitoring. *J. Field Robot.*, 34(8):1427–1449, 2017.

[63] Geoffrey A. Hollinger and Gaurav S. Sukhatme. Sampling-based robotic information gathering algorithms. *Int. J. Rob. Res.*, 33(9):1271–1287, aug 2014.

[64] Ronald A. Howard. *Dynamic programming and Markov processes.* John Wiley, 1960.

[65] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Grasping pomdps. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4685–4692. IEEE, 2007.

[66] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. Multi-agent online optimization with delays: Asynchronicity, adaptivity, and optimism. *arXiv preprint arXiv:2012.11579*, 2020.

[67] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[68] Jessica Hwang, Paulo Orenstein, Judah Cohen, Karl Pfeiffer, and Lester Mackey. Improving subseasonal forecasting in the western U.S. with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2325–2335, 2019.

[69] Michael Vavrousek Jakuba. *Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery.* PhD thesis, Massachusetts Institute of Technology, 2007.

[70] Syed Talha Jawaid and Stephen L Smith. Informative path planning as a maximum traveling salesman problem with submodular rewards. *Discr. Appl. Math.*, 186:112–127, 2015.

[71] AH Jazwinski. Stochastic process and filtering theory, academic press. *A subsidiary of Harcourt Brace Jovanovich Publishers*, 1970.

[72] Yuu Jinnai, Jee Won Park, Marlos C Machado, and George Konidaris. Exploration in reinforcement learning with deep covering options. In *International Conference on Learning Representations*, 2020.

[73] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999.

[74] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Online learning under delayed feedback. In *International Conference on Machine Learning*, pages 1453–1461, 2013.

[75] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[76] Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-) convex optimization: Optimism, composite objectives, and variational bounds. In *International Conference on Algorithmic Learning Theory*, pages 681–720. PMLR, 2017.

[77] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1/2):99–134, 1998.

[78] Carl L Kaiser, Dana R Yoerger, James C Kinsey, Sean Kelley, Andrew Billings, Justin Fujii, Stefano Suman, Michael Jakuba, Zachary Berkowitz, Christopher R German, et al. The design and 200 day per year operation of the autonomous underwater vehicle sentry. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 251–260. IEEE, 2016.

[79] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[80] Parameswaran Kamalaruban. Improved optimistic mirror descent for sparsity and curvature. *arXiv preprint arXiv:1609.02383*, 2016.

[81] Nare Karapetyan, Kelly Benson, Chris McKinney, Perouz Taslakian, and Ioannis Rekleitis. Efficient multi-robot coverage of a known environment. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1846–1852. IEEE, 2017.

[82] Jonguk Kim, Seung-Kyu Son, Dongsung Kim, Sang-Joon Pak, Ok Hwan Yu, Sharon L. Walker, Jihye Oh, Sun Ki Choi, Kongtae Ra, Youngtak Ko, Kyeong-Hong Kim, Jun-Ho Lee, and Juwon Son. Discovery of active hydrothermal vent fields along the Central Indian Ridge, 8–12° S. *Geochemistry, Geophysics, Geosystems*, 21(8), 2020.

[83] Genshiro Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[84] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

[85] Wouter Koolen, Tim Van Erven, and Peter Grunwald. Learning the learning rate for prediction with expert advice. In *Advances in Neural Information Processing Systems*, volume 27, pages 2294–2302, 2014.

[86] Alexander Korotin, Vladimir V'yugin, and Evgeny Burnaev. Adaptive hedging under delayed feedback. *Neurocomputing*, 397:356–368, 2020.

[87] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. *2005 Conf. Annu. Conf. Uncertain. Artif. Intell.*, pages 324–331, 2005.

[88] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

[89] Hanna Kurniawati and Vinay Yadav. An online POMDP solver for uncertainty planning in dynamic environment. In *Robotics Research*, pages 611–629. Berlin, Germany: Springer, 2016.

[90] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland., 2008.

[91] Tze Leung Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pages 1091–1114, 1987.

[92] Andrea L Lang, Kathleen Pegion, and Elizabeth A Barnes. Bridging weather and climate: Subseasonal-to-seasonal (S2S) prediction. *Journal of Geophysical Research: Atmospheres*, 125(4), 2020.

[93] Andrzej Lasota and Michael C. Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*, volume 97. Springer Science & Business Media, 1998.

[94] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms.* Cambridge University Press, 2020.

[95] Wee S Lee, Nan Rong, and David Hsu. What makes some POMDP problems easy to approximate? In *Advances in Neural Information Processing Systems*, pages 689–696, 2008.

[96] Martin Leutbecher and Tim N Palmer. Ensemble forecasting. *Journal of computational physics*, 227(7):3515–3539, 2008.

[97] Daniel Levine, Brandon Luders, and Jonathan How. Information-rich path planning with general constraints using rapidly-exploring random trees. In *AIAA Infotech at Aerospace 2010*, page 3360. AIAA, 2010.

[98] Laifang Li, Raymond W Schmitt, Caroline C Ummenhofer, and Kristopher B Karnauskas. Implications of North Atlantic sea surface salinity for summer precipitation over the US Midwest: Mechanisms and predictive value. *Journal of Climate*, 29(9): 3143–3159, 2016.

[99] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Monte Carlo Value Iteration with Macro-Actions. In *NIPS*, pages 1287–1295, 2011.

[100] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Adaptive informative path planning in metric spaces. *Int. J. Robot. Res.*, 35(5):585–598, 2016.

[101] Chun Kai Ling, Kian Hsiang Low, and Patrick Jaillet. Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. 30th AAAI conf. Artif. Intell.*, pages 1860–1866, 2016.

[102] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier, 1995.

[103] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian process meets big data: A review of scalable GPs. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423, 2020.

[104] Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

[105] Ji Liu, Steve Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. In *International Conference on Machine Learning*, pages 469–477. PMLR, 2014.

[106] Katherine Liu, Martina Stadler, and Nicholas Roy. Learned sampling distributions for efficient planning in hybrid geometric and object-level representations. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9555–9562. IEEE, 2020.

[107] E.N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20 (2):130–141, 1963.

[108] Sandeep Manjanna, Nikhil Kakodkar, Malika Meghjani, and Gregory Dudek. Efficient terrain driven coral coverage using Gaussian processes for mosaic synthesis. In *Proc. 13th IEEE Conf. Comput. Robot Vis.*, pages 448–455, 2016.

[109] Sandeep Manjanna, Alberto Quattrini Li, Ryan N Smith, Ioannis Rekleitis, and Gregory Dudek. Heterogeneous multi-robot system for exploration and strategic water sampling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4873–4880. IEEE, 2018.

[110] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18:50–60, 1947.

[111] Roman Marchant, Fabio Ramos, and Scott Sanner. Sequential Bayesian Optimisation for Spatial-Temporal Monitoring. In *Proc. 13th Conf. Uncertainty Artif. Intell.*, pages 553–562, 2014.

[112] James C Mason, Andrew Branch, Guangyu Xu, Michael V Jakuba, Christopher R German, Steve Chien, Andrew D Bowen, Kevin P Hand, and Jeffrey S Seewald. Evaluation of AUV Search Strategies for the Localization of Hydrothermal Venting. *Workshop on Planning and Robotics, International Conference on Automated Planning and Scheduling (ICAPS PlanRob 2020)*, 2020.

[113] Kathleen Mcgill and Stephen Taylor. Robot algorithms for localization of multiple emission sources. *ACM Computing Surveys (CSUR)*, 43(3):1–25, 2011.

[114] Brendan McMahan and Matthew Streeter. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, volume 27, pages 2915–2923, 2014.

[115] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

[116] Scott McQuade and Claire Monteleoni. Global climate model tracking using geospatial neighborhoods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 2012.

[117] William J. Merryfield, Johanna Baehr, Lauriane Batté, Emily J. Becker, Amy H. Butler, Caio A. S. Coelho, Gokhan Danabasoglu, Paul A. Dirmeyer, Francisco J. Doblas-Reyes, Daniela I. V. Domeisen, Laura Ferranti, Tatiana Ilynia, Arun Kumar, Wolfgang A. Müller, Michel Rixen, Andrew W. Robertson, Doug M. Smith, Yuhei Takaya, Matthias Tuma, Frederic Vitart, Christopher J. White, Mariano S. Alvarez, Constantin Ardilouze, Hannah Attard, Cory Baggett, Magdalena A. Balmaseda, Asmerom F. Beraki, Partha S. Bhattacharjee, Roberto Bilbao, Felipe M. de Andrade, Michael J. DeFlorio, Leandro B. Díaz, Muhammad Azhar Ehsan, Georgios Fragkoulidis, Sam Grainger, Benjamin W. Green, Momme C. Hell, Johnna M. Infanti, Katharina Isensee, Takahito Kataoka, Ben P. Kirtman, Nicholas P. Klingaman, June-Yi Lee, Kirsten Mayer, Roseanna McKay, Jennifer V. Mecking, Douglas E. Miller, Nele Neddermann, Ching Ho Justin Ng, Albert Ossó, Klaus Pankatz, Simon Peatman, Kathy Pegion, Judith Perlwitz, G. Cristina Recalde-Coronel, Annika Reintges, Christoph

Renkl, Balakrishnan Solaraju-Murali, Aaron Spring, Cristiana Stan, Y. Qiang Sun, Carly R. Tozer, Nicolas Vigaud, Steven Woolnough, and Stephen Yeager. Current and emerging developments in subseasonal to decadal prediction. *Bulletin of the American Meteorological Society*, 101(6):E869–E896, 2020.

[118] Chris Mesterharm. On-line learning with delayed label feedback. In *International Conference on Algorithmic Learning Theory*, pages 399–413. Springer, 2005.

[119] Mehryar Mohri and Scott Yang. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pages 848–856. PMLR, 2016.

[120] Claire Monteleoni and Tommi Jaakkola. Online learning of non-stationary sequences. In *Advances in Neural Information Processing Systems*, volume 16, pages 1093–1100, 2004.

[121] Claire Monteleoni, Gavin A Schmidt, Shailesh Saroha, and Eva Asplund. Tracking climate models. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(4):372–392, 2011.

[122] BR Morton, Geoffrey Ingram Taylor, and John Stewart Turner. Turbulent gravitational convection from maintained and instantaneous sources. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, pages 1–23, 1956.

[123] Soukayna Mouatadid, Paulo Orenstein, Genevieve Flaspohler, Miruna Oprescu, Judah Cohen, Franklyn Wang, Sean Knight, Maria Geogdzhayeva, Sam Levang, Ernest Fraenkel, and Lester Mackey. Learned benchmarks for subseasonal forecasting. *arXiv preprint arXiv:2109.10399*, 2021.

[124] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon markov decision process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

[125] Kentaro Nakamura, Tomohiro Toki, Nobutatsu Mochizuki, Miho Asada, Jun-ichiro Ishibashi, Yoshifumi Nogi, Shuro Yoshikawa, Jun-ichi Miyazaki, and Kyoko Okino. Discovery of a new hydrothermal vent based on an underwater, high-resolution geophysical survey. *Deep Sea Research Part I: Oceanographic Research Papers*, 74:1–10, 2013.

[126] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[127] Yu Nishiyama, Abdeslam Boularias, Arthur Gretton, and Kenji Fukumizu. Hilbert space embeddings of POMDPs. *arXiv preprint arXiv:1210.4887*, 2012.

[128] Ken Nowak, Jennifer Beardsley, Levi D Brekke, Ian Ferguson, and David Raff. Subseasonal Prediction for Water Management: Reclamation Forecast Rodeo I and II. In *100th American Meteorological Society Annual Meeting*. AMS, 2020.

[129] Francesco Orabona. A modern introduction to online learning. *ArXiv*, abs/1912.13213, 2019.

[130] Francesco Orabona and Dávid Pál. Scale-free algorithms for online linear optimization. In *International Conference on Algorithmic Learning Theory*, pages 287–301. Springer, 2015.

[131] Francesco Orabona and Dávid Pál. Optimal non-asymptotic lower bound on the minimax regret of learning with expert advice. *arXiv preprint arXiv:1511.02176*, 2015.

[132] Jason Pacheco and John Fisher. Variational information planning for sequential decision making. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2028–2036. PMLR, 2019.

[133] Jennifer B. Paduan, Robert A. Zierenberg, David A. Clague, Ronald M. Spelz, David W. Caress, Giancarlo Troni, Hans Thomas, Justin Glessner, Marvin D. Lilley, Thomas Lorenson, John Lupton, Florian Neumann, Miguel A. Santa Rosa-del Rio, and C. Geoffrey Wheat. Discovery of hydrothermal vent fields on Alarcón Rise and in Southern Pescadero Basin, Gulf of California. *Geochemistry, Geophysics, Geosystems*, 19(12):4788–4819, 2018.

[134] Georgios Papachristoudis and John W Fisher. Efficient information planning in Gaussian MRFs. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1–8. IEEE, 2015.

[135] Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[136] Kathy Pegion, Ben P Kirtman, Emily Becker, Dan C Collins, Emerson LaJoie, Robert Burgman, Ray Bell, Timothy DelSole, Dughong Min, Yuejian Zhu, et al. The Subseasonal Experiment (SubX): A multimodel subseasonal prediction experiment. *Bulletin of the American Meteorological Society*, 100(10):2043–2060, 2019.

[137] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research*, 27:335–380, 2006.

[138] Michael L Platt and Scott A Huettel. Risky business: the neuroeconomics of decision making under uncertainty. *Nature neuroscience*, 11(4):398–403, 2008.

[139] Robert Platt, Russell Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proc. Robot. Sci. Syst. Conf.*, 2010.

[140] Josep M Porta, Nikos Vlassis, Matthijs TJ Spaan, and Pascal Poupart. Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, 7 (Nov):2329–2367, 2006.

[141] Victoria Lynn Preston. *Adaptive sampling of transient environmental phenomena with autonomous mobile platforms*. Massachusetts Institute of Technology, 2019.

[142] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[143] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[144] Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In *Advances in Neural Information Processing Systems*, volume 28, pages 1270–1278, 2015.

[145] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1177–1184, 2008.

[146] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Proceedings of the 26th Annual Conference on Learning Theory*, pages 993–1019. PMLR, 2013.

[147] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[148] Carl E Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning.*, volume 14. MIT Press MIT Press, 2004.

[149] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 24, pages 693–701, 2011.

[150] R Tyrrell Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970.

[151] Bertrand Russell. *History of western philosophy.* New York: Simon and Schuster, 1967.

[152] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications.* PhD thesis, The Hebrew University, 2007.

[153] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[154] Florian Shkurti, Wei-Di Chang, Peter Henderson, Md Jahidul Islam, Juan Camilo Gamboa Higuera, Jimmy Li, Travis Manderson, Anqi Xu, Gregory Dudek, and Junaed Sattar. Underwater multi-robot convoying using visual tracking by detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4189–4196. IEEE, 2017.

[155] David Silver and Joel Veness. Monte Carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, pages 2164–2172, 2010.

[156] Amarjeet Singh, Andreas Krause, and William J Kaiser. Nonmyopic adaptive informative path planning for multiple robots. In *Proc. 21st Int. Joint Conf. Artif Intell.*, pages 1843–1850, 2009.

[157] Amarjeet Singh, Fabio Ramos, Hugh Durrant Whyte, and William J Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 5490–5497, 2010.

[158] W. D. Smart and L. Pack Kaelbling. Effective reinforcement learning for mobile robots. In *Proc. IEEE Int. Conf. Robot. Autom.*, volume 4, pages 3404–3410, 2002.

[159] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. DESPOT: Online POMDP planning with regularization. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1772–1780, 2013.

[160] Edward Jay Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, 1971.

[161] Christopher C Sotzing and David M Lane. Improving the coordination efficiency of limited-communication multi–autonomus underwater vehicle operations using a multiagent architecture. *Journal of Field Robotics*, 27(4):412–429, 2010.

[162] Matthijs TJ Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of artificial intelligence research*, 24:195–220, 2005.

[163] Kevin G Speer and Peter A Rona. A model of an Atlantic and Pacific hydrothermal plume. *Journal of Geophysical Research: Oceans*, 94(C5):6213–6220, 1989.

[164] Suvrit Sra, Adams Wei Yu, Mu Li, and Alex Smola. AdaDelay: Delay adaptive distributed stochastic optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 957–965. PMLR, 2016.

[165] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Trans. Inf. Theory*, 58:3250–3265, May 2012.

[166] Vishwak Srinivasan, Justin Khim, Arindam Banerjee, and Pradeep Ravikumar. Subseasonal Climate Prediction in the Western US using Bayesian Spatial Models. In *Uncertainty in artificial intelligence*, volume 37, 2021.

[167] Martina Stadler, Katherine Liu, and Nicholas Roy. Online high-level model estimation for efficient hierarchical robot navigation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5568–5575. IEEE, 2021.

[168] Gregory J Stein, Christopher Bradley, and Nicholas Roy. Learning over subgoals for efficient navigation of structured, unknown environments. In *Conference on Robot Learning*, pages 213–222. PMLR, 2018.

[169] Jacob Steinhardt and Percy Liang. Adaptivity and optimism: An improved exponentiated gradient algorithm. In *International Conference on Machine Learning*, pages 1593–1601, 2014.

[170] Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4): 354–356, 1969.

[171] Wen Sun, Niteesh Sood, Debadeepta Dey, Gireeja Ranade, Siddharth Prakash, and Ashish Kapoor. No-regret replanning under uncertainty. In *Proc. IEEE Inf. Conf. Robot. Autom.*, pages 6420–6427, 2017.

[172] Zachary N. Sunberg and Mykel J. Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. In *Proc. 24th Int. Conf. Automated Plan. Schedul.*, 2018.

[173] Zachary N Sunberg, Christopher J Ho, and Mykel J Kochenderfer. The value of inferring the internal state of traffic participants for autonomous freeway driving. In *Proc. IEEE Amer. Control Conf.*, pages 3004–3010, 2017.

[174] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[175] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[176] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E. Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[177] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas Hold'em. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[178] Georgios Theocharous and Leslie P Kaelbling. Approximate planning in POMDPs with macro-actions. In *Advances in Neural Information Processing Systems*, pages 775–782, 2004.

[179] Ali Tohidi and Nigel B Kaye. Highly buoyant bent-over plumes in a boundary layer. *Atmospheric Environment*, 131:97–114, 2016.

[180] A. Troccoli. Seasonal climate forecasting. *Meteorological Applications*, 17:251–268, 2010.

[181] Massimo Vergassola, Emmanuel Villermaux, and Boris I. Shraiman. Infotaxis as a strategy for searching without gradients. *Nature*, 445, Jan 2007.

[182] F Vitart, C Ardilouze, A Bonet, A Brookshaw, M Chen, C Codorean, M Déqué, L Ferranti, E Fucile, M Fuentes, et al. The subseasonal to seasonal (S2S) prediction project database. *Bulletin of the American Meteorological Society*, 98(1):163–173, 2017.

[183] Frédéric Vitart, Andrew W Robertson, and David LT Anderson. Subseasonal to seasonal prediction project: Bridging the gap between weather and climate. *Bulletin of the World Meteorological Organization*, 61(2):23, 2012.

[184] C Wang, Z Jia, Z Yin, F Liu, G Lu, and J Zheng. Improving the accuracy of subseasonal forecasting of china precipitation with a machine learning approach. front. *Earth Sci*, 9:659310, 2021.

[185] Lingxiao Wang, Shuo Pang, and Guangyu Xu. 3-dimensional hydrothermal vent localization based on chemical plume tracing. In *Global Oceans 2020: Singapore–US Gulf Coast*, pages 1–7. IEEE, 2020.

[186] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proc. 34th Int. Conf. Mach. Learn.*, volume 70, pages 3627–3635, 2017.

[187] Duncan Watson-Parris. Machine learning for weather and climate are worlds apart. *Philosophical Transactions of the Royal Society A*, 379(2194):20200098, 2021.

[188] Marcelo J Weinberger and Erik Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, 2002.

[189] Jonathan A Weyn, Dale R Durran, Rich Caruana, and Nathaniel Cresswell-Clay. Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *arXiv preprint arXiv:2102.05107*, 2021.

[190] Christopher J. White, Henrik Carlsen, Andrew W. Robertson, Richard J.T. Klein, Jeffrey K. Lazo, Arun Kumar, Frederic Vitart, Erin Coughlan de Perez, Andrea J. Ray, Virginia Murray, Sukaina Bharwani, Dave MacLeod, Rachel James, Lora Fleming, Andrew P. Morse, Bernd Eggen, Richard Graham, Erik Kjellström, Emily Becker, Kathleen V. Pegion, Neil J. Holbrook, Darryn McEvoy, Michael Depledge, Sarah Perkins-Kirkpatrick, Timothy J. Brown, Roger Street, Lindsey Jones, Tomas A. Remenyi, Indi Hodgson-Johnston, Carlo Buontempo, Rob Lamb, Holger Meinke, Berit Arheimer, and Stephen E. Zebiak. Potential applications of subseasonal-to-seasonal (S2S) predictions. *Meteorological applications*, 24(3):315–325, 2017.

[191] G Williams, Ted Maksym, Jeremy Wilkinson, Clayton Kunz, Chris Murphy, Peter Kimball, and Hanumant Singh. Thick and deformed Antarctic sea ice mapped with autonomous underwater vehicles. *Nature Geoscience*, 8(1):61–67, 2015.

[192] Jason L Williams. *Information theoretic sensor management.* PhD thesis, Massachusetts Institute of Technology, 2007.

[193] Jason L Williams, John W Fisher III, and Alan S Willsky. Performance guarantees for information theoretic active inference. In *Artificial Intelligence and Statistics*, pages 620–627, 2007.

[194] Akio Yamagami and Mio Matsueda. Subseasonal Forecast Skill for Weekly Mean Atmospheric Variability Over the Northern Hemisphere in Winter and Its Relationship to Midlatitude Teleconnections. *Geophysical Research Letters*, 47(17), 2020.

[195] N.K. Yilmaz, C. Evangelinos, P. Lermusiaux, and N.M. Patrikalakis. Path Planning of Autonomous Underwater Vehicles for Adaptive Sampling Using Mixed Integer Linear Programming. *IEEE J. Ocean. Eng.*, 33(4):522–537, 2008.

[196] Dana R. Yoerger, Molly Curran, Justin Fujii, Christopher R. German, Daniel Gomez-Ibanez, Annette F. Govindarajan, Jonathan C. Howland, Joel K. Llopiz, Peter H. Wiebe, Brett W. Hobson, Kakani Katija, Michael Risi, Bruce H. Robison, Cailean J. Wilkinson, Stephen M. Rock, and John A. Breier. Mesobot: An autonomous underwater vehicle for tracking and sampling midwater targets. In *2018 IEEE/OES Autonomous Underwater Vehicle workshop (AUV)*, pages 1–7. IEEE, 2018.

[197] Youtube. Buoyant Plume. https://www.youtube.com/watch?v=R3CmJDEPi_o, 2008. Accessed: 2022-05-17.

[198] Sue Zheng. *Accounting for Computational Expenditures in Bayesian Experimental Design*. PhD thesis, Massachusetts Institute of Technology, 2021.

[199] Sue Zheng, Jason Pacheco, and John Fisher. A robust approach to sequential information theoretic planning. In *International Conference on Machine Learning*, pages 5941–5949. PMLR, 2018.

[200] Sue Zheng, David Hayden, Jason Pacheco, and John W Fisher III. Sequential Bayesian experimental design with variable cost structure. *Advances in Neural Information Processing Systems*, 33:4127–4137, 2020.

[201] Martin Zinkevich, Michael Johanson, Michael H. Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, volume 20, 2007.