PROSPECTIVE LEARNING AND CONTROL

by
Ashwin De Silva

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
May 2026

Abstract

Learning involves updating decision rules based on past experience to improve future performance. The prevailing theoretical framework for studying machine learning, probably approximately correct (PAC) learning, assumes that future data will be drawn from the same fixed distribution as past data. This assumption is rarely satisfied in practice: data distributions drift over time, goals evolve, and the optimal hypothesis changes. Natural intelligences do not operate under this assumption. Biological systems, from mitochondria anticipating muscular demand, to neural circuits predicting sensory input, to animals timing their departure from a depleting food patch, prospectively model how the world changes. Rather than reacting to change, they act in anticipation of it.

This thesis develops a formal framework, called prospective learning, that brings this prospective capacity into statistical learning theory. The framework makes three modifications to the classical setup: data is modeled as a stochastic process rather than independent draws from a fixed distribution; the hypothesis is a time-indexed sequence of predictors, or equivalently a function that takes time as an input alongside the data; and risk is defined as a conditional expectation of cumulative future loss given the realized past.

The thesis makes four contributions. First, it shows that even in the classical two-distribution setting, target generalization error is a non-monotonic function of the amount of non-current data, a phenomenon not predicted by existing theory. It then

characterizes when and how non-current data can be optimally exploited. Second, it proves that time-agnostic empirical risk minimization provably cannot achieve Bayes risk for certain stochastic processes. By contrast, prospective ERM, which differs only in that time is provided as an additional input, is a strong prospective learner under consistency and uniform concentration conditions. Third, it extends the framework to sequential decision-making in non-stationary, single-life environments without requiring the Markov decision process assumption, and introduces a prospective control algorithm that achieves near-oracle performance orders of magnitude faster than reinforcement learning baselines. Fourth, it shows that PAC learning, domain adaptation, meta-learning, continual learning, online learning, and reinforcement learning are all special cases of the prospective learning framework, corresponding to specific assumptions about the temporal structure of the data and the learner's capacity to act.

The central message of the thesis is that time is cheap: making time an input to a hypothesis is a trivial modification to any learning system, but it transforms what the system can achieve. A time-agnostic learner can adapt to the present; a time-aware learner can anticipate the future.

Keywords: prospective learning, distribution shift, stochastic processes, empirical risk minimization, prospective control, non-stationary environments, time-indexed hypotheses

Primary reader and thesis advisor

Dr. Joshua T. Vogelstein
Associate Professor
Department of Biomedical Engineering
Johns Hopkins University, Baltimore MD

Secondary readers

Dr. Carey E. Priebe
Professor
Department of Applied Mathematics and Statistics
Johns Hopkins University, Baltimore, MD

Dr. Pratik Chaudhari
Assistant Professor
Department of Electrical and Systems Engineering
University of Pennsylvania, Philadelphia, PA

To Malsha, for being perfect

Acknowledgement

This dissertation marks the culmination of a journey that has spanned over twenty-five years of continuous learning. It is as much a reflection of the people who shaped me as it is of the work itself. I am deeply grateful to everyone who, in ways both large and small, made this journey possible.

I begin with my family, to whom I owe everything. To *Ammi*, *Thaththi*, *Nangi*, *Seeya*, *Achchi-Amma*, *Punchi*, and Priyankara Uncle: thank you for raising me with care, patience, and unwavering support. Your sacrifices and belief in me have been the foundation upon which all of this rests.

To my wife, Malsha: thank you for being the constant light and strength in my life. Your love, resilience, and belief in me carried me through the most challenging moments of this journey. We were married in August 2021 and, just a week later, flew to the United States to begin both of our PhDs amidst a global pandemic. It is a leap that defined so much of this chapter of our lives. I am especially grateful for the many thoughtful and enriching conversations we shared along the way, which shaped this work in meaningful ways. This accomplishment is as much yours as it is mine.

I would also like to express my deepest gratitude to the people of Sri Lanka, whose commitment to free education made my journey possible from primary school through my undergraduate studies. I have been profoundly fortunate to benefit from this system, and I remain deeply aware that this opportunity carries with it a responsibility to contribute meaningfully in return.

I am deeply grateful to Dr. Janaka Senarathna (*Janaka Ayya*) for helping us settle in Baltimore, for being a lifelong friend, and for the many thoughtful and stimulating conversations that have enriched my perspective over the years.

My sincere thanks go to my teachers at Richmond College, where my curiosity first took root. I am equally grateful to the lecturers at the University of Moratuwa, particularly within the Department of Electronic and Telecommunication Engineering (ENTC), where I built my foundations in signals and systems, control theory, and electronics. Those formative years shaped the way I think about systems and learning.

I would like to warmly acknowledge Prof. Saman Halgamuge and Steven Petrou for the opportunity to intern under their guidance at the University of Melbourne and the Florey Institute of Neuroscience and Mental Health. It was there that I first discovered the excitement and joy of research, an experience that planted the seed that ultimately led me here.

I am also grateful to Dr. Simon Kappel, Dr. Nuwan Dayananda, and Dr. Anjula De Silva, whose guidance during my time at the ENTC played an important role in shaping my academic direction.

To *Udaranga Ayya*, *Adhitha Ayya*, Dr. Tharindu Perera, and Dr. Ransalu Senanayake: thank you for mentoring me through the years. Your advice helped me navigate what initially felt like an overwhelming path.

I owe a profound debt of gratitude to my advisor, Prof. Joshua T. Vogelstein. Jovo, thank you for being not only an exceptional advisor but also a deeply thoughtful and

kind human being. Your guidance and your insistence on thinking critically have shaped me in lasting ways. You also encouraged me to take many excellent courses across Johns Hopkins University, which broadened my perspective and played an important role in shaping this work. You set a high bar, not just for research, but for how to approach problems with clarity and purpose.

My sincere thanks to Prof. Pratik Chaudhari, my co-advisor. Our conversations, always insightful and rigorous, taught me the importance of precision in both thought and expression. Our collaboration has been a true highlight of my time in graduate school, and I learned a great deal from working with you. I am also grateful for the opportunities you created by inviting me to have discussions at UPenn, which were both enriching and enjoyable. I am deeply grateful for your guidance and mentorship.

I am equally grateful to Prof. Carey E. Priebe, who introduced me to statistics and learning theory in a way that was both elegant and profound. A single, beautifully written one-page problem you posed became the seed for my first project, and ultimately, for much of the work in this dissertation.

I would also like to thank Dr. Adam Charles, Dr. Avanti Athreya, Dr. Jeremias Sulam, Dr. Marlos C. Machado, and Dr. Alex Robey for their constructive feedback and stimulating discussions, which helped sharpen many aspects of this work.

I am grateful to my course professors across the Johns Hopkins University Whiting School of Engineering and the School of Medicine, from whom I gathered a wealth of knowledge that became instrumental in shaping this dissertation.

Acknowledgement

I would like to extend my sincere thanks to Vatshank, my internship manager at Amazon, for being an exceptional mentor. I learned a great deal under your guidance, and your support had a lasting impact on my development.

To Rahul, thank you for being an incredible collaborator, a mentor and a friend. Our frequent discussions (sometimes even at late night!), where no question was too small or too large, were some of the most rewarding moments of this journey.

I am deeply grateful to Rubing and Aranyak, whose mathematical insight and persistence led to some of the most challenging proofs in this dissertation.

To Hayden, thank you for being an amazing collaborator. It was a pleasure working together, and I truly valued our many thoughtful discussions along the way.

To Cecelia and Yuxin, thank you for the opportunity to work together on prospective control and foraging. I am grateful for the chance to learn alongside you.

I would also like to thank the past and present members of the NeuroData community, including Ben, Tommy, Jayanta, Suki, Hao, Skyler, Eric, and Mike. Being part of this group was a highlight during my time at Johns Hopkins University.

My sincere thanks to Kay, Kim, and Anthony for their unwavering administrative support. Your help behind the scenes made so many aspects of this journey smoother than they otherwise would have been.

Finally, to Harindi, Joanne, Shanuki, Navindi, and Praga: thank you for making my time in Baltimore and beyond so memorable. Your friendship brought balance, joy,

and a sense of home during these years. I would also like to extend my heartfelt gratitude to Sasith, Yomindu, Dumindu, and Pasindu, for their enduring friendship and support across the distance.

To everyone mentioned here, and to those I may have unintentionally missed: thank you. This dissertation stands as a testament not only to my efforts, but to the collective support, mentorship, and friendship that made it possible.

*"...the unrivaled human ability to be guided by imagining alternatives stretching into the future—**prospection**—uniquely describes Homo sapiens."*

– Martin E. P. Seligman, *Homo Prospectus*

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Prospection in natural intelligence

Living systems anticipate the future. This capacity, *prospection*, is not a peripheral feature of biological cognition but a central organizing principle, present across scales from cellular regulation to high-level planning.

At the cellular level, mitochondria increase energy production in advance of muscular demand, not in response to it. The regulatory principle governing this behavior is allostasis (Sterling, 2012): the body's internal systems predict the organism's needs and prepare to satisfy them *before* they arise, rather than correcting deviations after the fact. Allostasis is distinct from homeostasis, which maintains equilibrium by reacting to perturbations. The allostatic system acts prospectively; the homeostatic system acts retrospectively.

At the neural level, predictive coding (Friston and Kiebel, 2009; Huang and R. P. N. Rao, 2011; Y. Song et al., 2024), the hypothesis that the brain maintains generative models that predict incoming sensory signals and transmit only the prediction errors, implies that perception itself is a prospective act. The brain does not passively receive stimuli; it anticipates them, and updates its models only when the anticipation is wrong. This anticipatory architecture is metabolically efficient (prediction errors are sparser than raw signals) and computationally powerful (it allows the brain to fill in

occluded objects, track moving targets, and prepare motor responses before a stimulus arrives).

At the behavioral level, humans engage in *mental time travel* (Suddendorf et al., 2009; M. E. Seligman et al., 2013; M. E. P. Seligman et al., 2016; Raby and Clayton, 2009; Kording et al., 2025; Vogelstein, Verstynen, et al., 2022): the ability to project oneself into future scenarios, simulate their consequences, and choose actions accordingly. This capacity is not limited to explicit planning. Prospective memory (McDaniel and Einstein, 2007), the ability to remember to perform an intended action at an appropriate future moment, is a routine cognitive operation that requires maintaining a representation of "what to do when". Foraging animals, navigating between food patches whose yields fluctuate over time, must decide when to leave a depleting patch and travel to an alternative, accepting zero immediate reward during transit in anticipation of future gain (Charnov, 1976; Pyke, 1984). This departure-timing problem requires a model of how rewards evolve over time, a model that is, in the language of this thesis, a time-indexed hypothesis.

A related capacity is reversal learning (Izquierdo et al., 2017; Costa et al., 2015), the ability to detect and adapt when previously learned contingencies are reversed. Reversal learning experiments, in which the rewarded and unrewarded stimuli swap roles, are a standard assay for cognitive flexibility across species from insects to primates. This thesis formalizes two variants of this paradigm. The alternating-tasks process of Sections 2.4 and 3.3, in which two classification distributions with flipped labels arrive in a predictable temporal pattern, captures the perceptual side of reversal learning: the learner must anticipate which contingency is active at each moment.

The prospective foraging task of Section 3.7 and Chapter 4, in which two reward patches alternate in a periodic schedule, captures the active side: the agent must not only anticipate the reversal but act on that anticipation, departing the current patch before its reward is depleted and arriving at the alternative in time to exploit the new contingency. In both cases, a retrospective learner that merely detects the reversal after it occurs is structurally outperformed by a prospective learner that anticipates it.

The breadth of these examples suggests that prospection is not a specialized cognitive module but a general design principle: biological systems that persist through time build models of how the world changes and use those models to act in advance of change. The question this thesis asks is whether the same principle can be formalized and made useful in machine learning.

1.2 The retrospective bias of machine learning

Modern machine learning is built on a foundation that ignores time. The probably approximately correct (PAC) learning framework (Vapnik, 1998; L. Valiant, 2013; L. G. Valiant, 1984; Vapnik, 1991) assumes that training and test data are drawn independently from the same fixed distribution (Glivenko, 1933; Valery Glivenko, 1933). Empirical risk minimization (ERM), the workhorse algorithm, selects a hypothesis that performs well on past data and deploys it, unchanged, on future data. The implicit assumption is that the future will look like the past.

This assumption has been extraordinarily productive. It underlies the successes of deep learning in computer vision (Krizhevsky, 2009; He, X. Zhang, et al., 2016; He,

Gkioxari, et al., 2017; Ho et al., 2020; Radford, Kim, Hallacy, et al., 2021), language modeling (Mikolov et al., 2013; Vaswani et al., 2017; Radford, Narasimhan, et al., 2018; Guo et al., 2025), protein structure prediction (Jumper et al., 2021; Senior et al., 2020), audio and speech processing (Radford, Kim, T. Xu, et al., 2023; Schneider et al., 2019), and many other domains. But it is neither testable nor believed to be true in practice. Real data distributions drift over time: user preferences evolve, sensor characteristics degrade, environmental conditions change (Quinonero-Candela et al., 2008; Koh et al., 2021; Yao et al., 2022; Žliobaitė et al., 2015; Ben-David and Urner, 2012; Mohri and Muñoz Medina, 2012). When the future differs from the past, models trained under the IID assumption can fail silently, producing confident but wrong predictions.

The machine learning community has developed numerous strategies to address this gap. Covariate shift methods (Sugiyama et al., 2008; Reddi et al., 2015; Ben-David and Urner, 2012) reweight training data to match the test distribution. Domain adaptation (Ben-David, Blitzer, et al., 2010; Mansour et al., 2008; Pan et al., 2010; Ganin et al., 2016; Cortes et al., 2019) maps representations across domains. Meta-learning (Finn, Abbeel, et al., 2017; Maurer and Jaakkola, 2005; Baxter, 2000; Romera-Paredes and Torr, 2015; Snell et al., 2017; Finn, Rajeswaran, et al., 2019) trains models that can adapt quickly to new tasks. Continual learning (R. Ramesh and Chaudhari, 2021; Vogelstein, Dey, et al., 2020; Thrun, 1998; Van de Ven and Tolias, 2019; De Lange and Tuytelaars, 2021; Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017) seeks to retain performance on old tasks while learning new ones. Online learning (Shalev-Shwartz, 2025; Mohri, Rostamizadeh, et al., 2018; Nicolò

Cesa-Bianchi and Orabona, 2021) makes no distributional assumptions and minimizes worst-case regret. Reinforcement learning (Sutton and Barto, 1998; Strehl et al., 2009; Levine et al., 2020) handles sequential decision-making in stationary, Markov environments with episodic resets.

Each of these strategies makes a specific assumption about the relationship between past and future data. Covariate shift assumes the marginal on inputs changes but the conditional on labels does not. Meta-learning assumes tasks are drawn IID from a meta-distribution. Continual learning assumes the learner knows (or can detect) when the task changes. Reinforcement learning assumes the Markov property, stationarity, and the availability of resets.

None of these strategies does what biological prospection does: model how the world changes over time and use that model to anticipate the future (M. E. Seligman et al., 2013; Raby and Clayton, 2009). A meta-learner cannot predict which task comes next, because it assumes tasks are exchangeable. A continual learner adapts to new tasks after they arrive, but does not anticipate them before they arrive. An online learner optimizes against the worst case, not against the structure of the temporal dynamics. Each strategy is retrospective in a precise sense: it uses past data to build a hypothesis that is optimal for the past (De Silva, Ramesh, and Yang et al., 2024; De Silva and Ramesh et al., 2023b), and deploys that hypothesis on the future, possibly adapted and possibly regularized, but fundamentally unchanged in its relationship to time.

1.3 Thesis question

This thesis asks:

"Under what conditions can a learner, using only data from the past, make predictions that remain accurate indefinitely into a future whose distribution evolves over time — and how must the learner be designed to achieve this?"

The answer developed in the following chapters is that the learner must make time a first-class variable: the data must be modeled as a stochastic process, the hypothesis must be a function of time, and the risk must be defined as a conditional expectation of cumulative future loss given the realized past. Under consistency and concentration conditions analogous to those of PAC learning, ERM over such time-indexed hypotheses achieves Bayes risk. Without time as an input, ERM provably cannot.

1.4 Unifying notation

The thesis uses a consistent notation across all chapters. The core objects are defined here and used without re-definition in what follows.

Data. The data is a stochastic process $Z = (Z_t)_{t\in\mathbb{N}}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with each $Z_t = (X_t, Y_t)$ taking values in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Here $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. The process generates a filtration $\mathcal{F}_t = \sigma(Z_1, \ldots, Z_t)$. The realized past is denoted $z_{\leq t} = (z_1, \ldots, z_t)$; the corresponding random variable is $Z_{\leq t}$.

Hypothesis. A hypothesis sequence is $h = (h_1, h_2, \ldots)$ with each $h_t : \mathcal{X} \to \mathcal{Y}$, or equivalently a function $h : \mathbb{N} \times \mathcal{X} \to \mathcal{Y}$. A hypothesis class is $\mathcal{H} \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$. A time-agnostic hypothesis satisfies $h_t = h_{t'}$ for all t, t'.

Loss and risk. The per-step loss is $\ell : \mathbb{N} \times \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. The prospective loss is $\bar{\ell}_t(h, Z) = \limsup_{\tau \to \infty} \frac{1}{\tau} \sum_{k=t+1}^{t+\tau} \ell(k, h_k(X_k), Y_k)$. The prospective risk is $R_t(h) = \mathbb{E}[\bar{\ell}_t(h, Z) \mid Z_{\leq t}]$. The Bayes risk is $R_t^* = \inf_{h \in \mathcal{F}_t} R_t(h)$.

Special cases. Chapter 2 uses the two-distribution special case: $\mathcal{T} = \{t, o\}$, target distribution P_t, OOD distribution P_o, n target samples, m OOD samples, target generalization error $e_t(h) = \mathbb{E}_{(x,y)\sim P_t}[\mathbb{1}\{h(x) \neq y\}]$. Chapter 4 extends to the control setting: state space $\mathcal{S}$, action space $\mathcal{A}$, hypothesis $h : \mathcal{S} \times \mathbb{N} \to \mathcal{A}$, reward $r_t(s)$.

1.5 Contributions and roadmap

The thesis makes four contributions, developed across four chapters.

Contribution 1: A characterization of when non-current data helps (Chapter 2). The target generalization error of a learner trained on a mixture of target and OOD data is non-monotonic in the amount of OOD data. This phenomenon is demonstrated analytically via Fisher's Linear Discriminant and empirically via deep networks on MNIST, CIFAR-10, PACS (Li, Y. Yang, Y.-Z. Song, and Timothy M. Hospedales, 2017b), and DomainNet (Gulrajani and Lopez-Paz, 2020). It is not predicted by existing generalization bounds and is not resolved by standard remedies. In an oracle setting where OOD sample identities are known, a weighted-ERM procedure with

an appropriately chosen α rectifies the non-monotonicity. For a Gaussian generative model with von Mises–Fisher structure on projection vectors, the optimal α^* can be computed in closed form. This chapter is based on De Silva and Ramesh et al. (2023a) and Helm and De Silva et al. (2024).

Contribution 2: A formal framework for prospective learning (Chapter 3). Data is modeled as a stochastic process; the hypothesis is a time-indexed sequence of predictors; risk is defined as a conditional expectation of time-averaged future loss. Time-agnostic ERM provably cannot achieve Bayes risk for certain stochastic processes (Proposition 3.3.1). Prospective ERM, which differs from time-agnostic ERM only in that time is provided as an additional input, is a strong prospective learner under consistency and uniform concentration conditions (Theorem 3.4.1). A countable hypothesis class suffices (Theorem 3.4.2). The framework is validated on synthetic, MNIST, and CIFAR-10 tasks across multiple scenarios and is extended to decision-forest-based learners. This chapter is based on De Silva, Ramesh, and Yang et al. (2024), De Silva and Ramesh et al. (2023b), Bai, Shuai and De Silva et al. (2026), and Bai, Acharyya, and De Silva et al. (2025).

Contribution 3: An extension to prospective control (Chapter 4). The framework is extended to sequential decision-making in non-stationary, single-life environments without the MDP assumption. A learnability theorem analogous to Theorem 4.3.1 is stated. The PLC algorithm, combining instantaneous and cumulative loss regressors with short-horizon planning, achieves near-oracle performance on a prospective foraging task, converging in approximately 100 time steps where time-aware RL baselines require 10^3 to 10^5. This chapter is based on Bai, Acharyya, and

De Silva et al. (2025).

Contribution 4: A unified view of learning paradigms (Chapter 5). PAC learning, domain adaptation, meta-learning, continual learning, online learning, and reinforcement learning are all shown to be special cases of the prospective learning framework, corresponding to specific assumptions about the temporal structure of the stochastic process and the learner's capacity to act. Several open problems are identified, including complexity measures for stochastic processes, finite-sample rates, connections to adaptive control and computational mechanics, and the time-embedding-selection problem.

1.6 Related work

This section provides a brief orientation to the literature most relevant to the thesis. Detailed technical comparisons are deferred to the individual chapters.

Distribution shift and domain adaptation. The classical framework for learning under distribution shift models the discrepancy between source and target distributions and provides generalization bounds as a function of this discrepancy. The foundational result is the bound of Ben-David, Blitzer, et al. (2010), which the thesis discusses in Section 2.1. Domain adaptation methods (Mansour et al., 2008; Pan et al., 2010; Ganin et al., 2016; Cortes et al., 2019; Gulrajani and Lopez-Paz, 2020; Arjovsky et al., 2019; Sun and Saenko, 2016) — including importance weighting, representation alignment, and domain-adversarial training — seek to correct for the shift, typically assuming a single, known shift between source and target. The thesis's Chapter 2

extends this literature by showing that the target generalization error can be non-monotonic in the amount of source data, a phenomenon the Ben-David bound does not predict.

Multi-task, meta-, and continual learning. Multi-task learning (Baxter, 2000) trains a shared representation across tasks. Meta-learning (Finn, Abbeel, et al., 2017; Maurer and Jaakkola, 2005; Finn, Rajeswaran, et al., 2019) trains a learning algorithm that can adapt to new tasks from few examples. Continual learning (R. Ramesh and Chaudhari, 2021; Vogelstein, Dey, et al., 2020; Thrun, 1998) addresses sequences of tasks without forgetting previous ones. All three assume that the task identity is known or that tasks are exchangeable. Prospective learning differs in that it models the temporal dynamics of how tasks evolve and uses these dynamics to anticipate future tasks rather than adapt to them retrospectively.

Online learning. Online learning (Shalev-Shwartz, 2025; Mohri, Rostamizadeh, et al., 2018; Nicolò Cesa-Bianchi and Orabona, 2021) makes no distributional assumptions and evaluates performance via regret, the gap between the learner's cumulative loss and that of the best fixed hypothesis in hindsight. Prospective learning makes distributional assumptions (the data is a stochastic process with exploitable structure) and evaluates performance via the gap to the Bayes-optimal sequence of hypotheses. The two frameworks are complementary: online learning provides worst-case guarantees when no structure is available; prospective learning provides stronger guarantees when structure is available.

Reinforcement learning. RL (Sutton and Barto, 1998; Strehl et al., 2009; Levine et al., 2020) addresses sequential decision-making under the MDP assumption with stationary dynamics and episodic resets. Prospective control (Chapter 4) relaxes all three assumptions. Recent work on continual RL (S. Kumar, Marklund, A. Rao, Y. Zhu, Jeon, Yueyang, et al., 2025; Khetarpal et al., 2022), single-life RL (Chen et al., 2022), and non-stationary RL moves toward similar relaxations from the RL side; Chapter 5 discusses the relationship between these lines of work and PLC.

Forecasting and time-series analysis. Forecasting (Petropoulos et al., 2022; Ghosh and Sen, 1991; Nicolo Cesa-Bianchi and Lugosi, 2006) assumes data from a stochastic process and predicts future values over a fixed or rolling horizon. Prospective learning differs in that the learner predicts over the infinite future, the loss can be time-varying, and the hypothesis is a classifier or decision-maker rather than a point forecast. Probabilistic forecasting (Gneiting and Katzfuss, 2014), which maintains and iteratively updates a distribution over future trajectories, is the closest relative; the connection is discussed in Chapter 5.

Information theory and computational mechanics. Predictive information (Bialek et al., 2001), the information bottleneck (Tishby et al., 2000), and ϵ-machines (Shalizi and Crutchfield, 2001) provide information-theoretic characterizations of the structure in stochastic processes that enables prediction. These quantities are natural candidates for a prospective complexity measure analogous to VC dimension (Vapnik, 1998). The connections are discussed in Chapter 5 as open problems.

Chapter 2

Learning from Non-Current Data

The standard assumption in statistical learning, that training and test data are drawn independently from the same distribution, is rarely satisfied in practice. Data collected at different times, from different devices, or under different experimental conditions may follow distributions that differ from the one the learner will face at deployment. The classical response to this observation is domain adaptation: model the discrepancy between source (non-current) and target (current) distributions, and correct for it.

This chapter asks a more basic question: *when does non-current data help at all?* The answer turns out to be surprising. Adding data from a different distribution can improve generalization on the target up to a point. Beyond that point, the same data makes things worse. Target generalization error is a *non-monotonic function* of the amount of non-current data. This phenomenon is not predicted by existing generalization bounds, and it is not resolved by standard remedies such as data augmentation, hyperparameter optimization, or pretraining.

The chapter presents two complementary perspectives on this phenomenon. Section 2.2 establishes it broadly: analytically via Fisher's Linear Discriminant (Devroye et al., 1997) on synthetic data, and empirically via deep networks on MNIST, CIFAR-10, PACS, and DomainNet. Section 2.3 provides a mechanistic account: for a class of linear models with a specific generative structure, the optimal combination of source and target information can be computed in closed form, revealing the bias–variance

tradeoff in task space that underlies the non-monotonicity. Section 2.4 identifies the structural limitation of the two-distribution framing and motivates the transition to Chapter 3.

The chapter uses the following notation. A distribution P is a joint distribution over inputs $\mathcal{X}$ and outputs $\mathcal{Y}$. The target distribution is P_t; the non-current (out-of-distribution, or OOD) distribution is P_o. The learner has access to n samples from P_t and m samples from P_o. The goal is to minimize the generalization error $e_t(h) = \mathbb{E}_{(x,y)\sim P_t}[\mathbb{1}\{h(x) \neq y\}]$ on the target distribution.

In the notation of Chapter 1, this is the special case where the index set $\mathcal{T}$ has two elements, one for the target distribution and one for the source, and the learner must produce a single hypothesis rather than a time-indexed sequence. The chapter's concluding section will show why this special case, however carefully analyzed, cannot extend to the setting where $\mathcal{T} = \mathbb{N}$.

2.1 Setup and background

Consider a binary classification problem. The target distribution P_t and the OOD distribution P_o are both joint distributions over $\mathcal{X} \times \{0, 1\}$. The learner has access to a combined dataset of n target samples and m OOD samples, and seeks a hypothesis $h : \mathcal{X} \to \{0, 1\}$ that minimizes the target generalization error $e_t(h)$.

If the learner is unaware that some samples are OOD, the *OOD-agnostic* setting, it treats all $n + m$ samples as if they came from a single distribution and minimizes the pooled empirical risk:

$$\hat{e}(h) = \frac{1}{n+m} \sum_{i=1}^{n+m} \ell(h(x_i), y_i). \tag{2.1}$$

When $P_t = P_o$, this is standard ERM and the generalization error decreases as $O((n+m)^{-1/2})$. When $P_t \neq P_o$, the situation is less clear. Intuitively, if the two distributions are similar, additional OOD samples should help by reducing variance; if they are dissimilar, the OOD samples should hurt by introducing bias. One might therefore expect the generalization error to be monotonic in m: either decreasing (similar distributions) or increasing (dissimilar distributions).

The main finding of Section 2.2 is that this expectation is wrong. There is a third possibility: the error initially decreases with small amounts of OOD data and then increases with larger amounts. Generalization error is non-monotonic in m.

The most relevant prior theory for this setting is the generalization bound of Ben-David, Blitzer, et al. (2010). For a hypothesis $\hat{h}_\alpha$ that minimizes the α-weighted empirical loss $\alpha\hat{e}_t(h) + (1-\alpha)\hat{e}_o(h)$, the target error is bounded by

$$e_t(\hat{h}_\alpha) \leq e_t(h_t^*) + 4\sqrt{\left(\frac{\alpha^2}{n} + \frac{(1-\alpha)^2}{m}\right)\sqrt{V_{\mathcal{H}} - \log\delta}} + 2(1-\alpha)d_{\mathcal{H}}(P_t, P_o),$$

with probability at least $1-\delta$, where h_t^* is the target-risk minimizer, $V_{\mathcal{H}}$ is proportional to the VC dimension, and $d_{\mathcal{H}}(P_t, P_o)$ is a distributional divergence. This bound is monotonic in m for fixed α and therefore *cannot* predict the non-monotonic phenomenon (see Section A.6).

2.2 The value of out-of-distribution data

2.2.1 Non-monotonicity in a tractable model

Consider a one-dimensional binary classification problem. Target samples are drawn from a Gaussian mixture with class-conditional means $\{-\mu, +\mu\}$ and common variance σ^2. OOD samples are drawn from a translated mixture with means $\{-\mu+\Delta, +\mu+\Delta\}$ and the same variance. The parameter Δ controls the dissimilarity between the two distributions: when $\Delta = 0$ the distributions are identical; as Δ grows, the OOD class means shift away from the target's. The Fig. 2.1 depicts the two distributions pictorially.

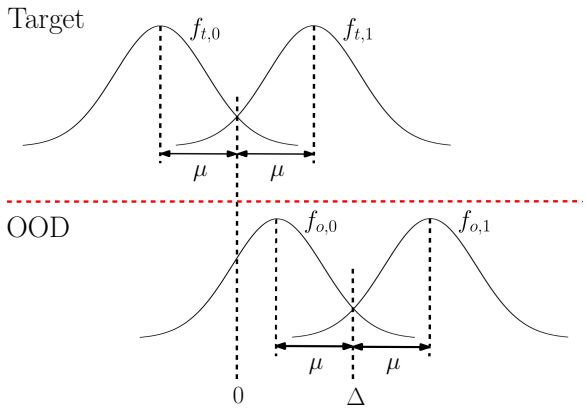


Figure 2.1: A picture of the target and non-current (or OOD) distributions.

Fisher's Linear Discriminant (FLD) is the Bayes-optimal classifier for Gaussian class-conditionals with shared covariance. For equal class priors, FLD reduces to thresholding at the midpoint of the estimated class means: $\hat{h}(x) = \mathbb{1}\{x > (\hat{\mu}_0 + \hat{\mu}_1)/2\}$ (see Section A.1). When the learner pools target and OOD samples without knowing which is which, the estimated class means are shifted by the OOD data, and the decision threshold is displaced from its optimal value.

The expected target generalization error of this pooled FLD has a closed-form expression:

$$e_t(\hat{h}) = \frac{1}{2}\left[\Phi\left(\frac{m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right) + \Phi\left(\frac{-m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right)\right], \tag{2.2}$$

where Φ is the standard normal CDF (see Section A.2 for the derivation). This expression reveals the non-monotonic behavior directly as illustrated in Fig. 2.2. For small m, the additional samples reduce the variance of the threshold estimate, decreasing the error. For large m, the OOD-induced bias in the threshold dominates, increasing the error. The transition between the two regimes depends on Δ: for small Δ (similar distributions), the error decreases monotonically; for large Δ (dissimilar distributions), the error first decreases, reaches a minimum at some optimal m^*/n, and then increases.

The mechanism is a bias–variance tradeoff in the space of decision thresholds (see Fig. 2.3). The mean squared error of the threshold decomposes as $\mathrm{MSE}(\hat{c}) = \mathrm{Bias}^2(\hat{c}) + \mathrm{Var}(\hat{c})$. The variance decreases monotonically with m, meaning more data always reduces estimation noise. The squared bias increases monotonically with m, meaning more OOD data pulls the threshold further from the target optimum. Together, their sum is non-monotonic.

This result is stated formally:

Theorem 2.2.1. There exist target and OOD distributions P_t and P_o such that

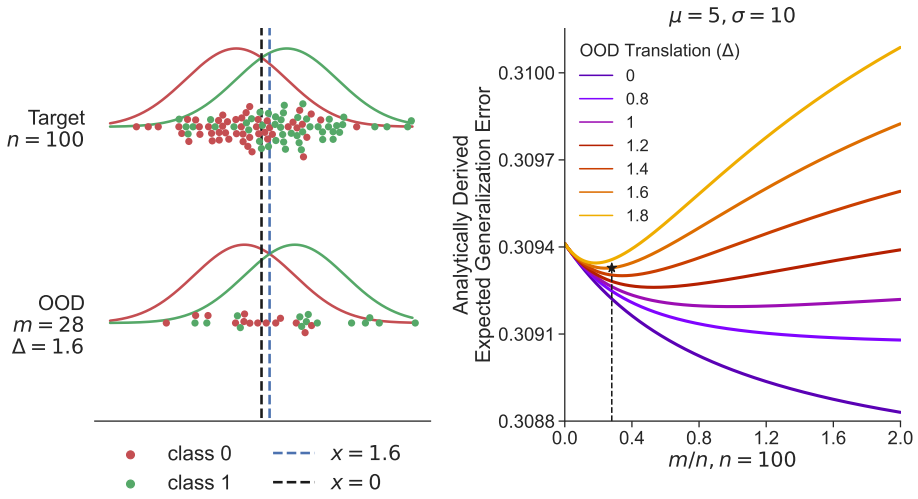


Figure 2.2: Left: A schematic of the Gaussian mixture model corresponding to the target (top) and OOD samples (bottom). The OOD sample size ($m = 28$) at which the target generalization error is minimized at $\Delta = 1.6$ is indicated at the top. **Right:** For $n = 100$, we plot the expected target generalization error of FLD as a function of the ratio of OOD and target samples m/n, for different types of OOD samples corresponding to different values of Δ. This plot uses the analytical expression for the generalization error in Eq. (2.2). For small values of Δ, when the two distributions are similar to each other, the generalization error decreases monotonically. However, beyond a certain value of Δ, the generalization error is non-monotonic in the number of OOD samples. The optimal value of m/n which leads to the best generalization error is a function of the relatedness between the two distributions, as governed by Δ in this example.

the target generalization error of the hypothesis minimizing the pooled empirical loss is non-monotonic in the number of OOD samples m. In particular, there exist distributions for which the error decreases with small amounts of OOD data and increases with larger amounts, relative to using no OOD data at all.

The FLD example is the constructive proof.

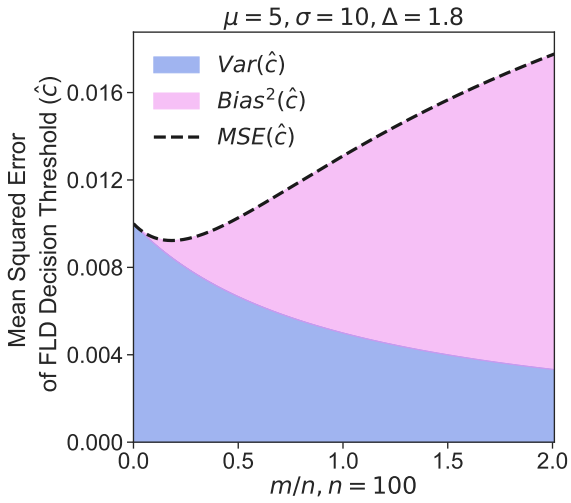


Figure 2.3: Mean squared error (Y-axis) of the decision threshold $\hat{c}$ of FLD (see Section A.2), for the same setup as that of Fig. 2.1, plotted against the ratio of the OOD and target samples m/n (X-axis) for $\Delta = 1.8$. Squared bias and variance of the MSE are in violet and blue, respectively. This illustration clearly demonstrates the intuition behind non-monotonic target error: the MSE drops initially because of the smaller variance due to the OOD samples. With more OOD samples, MSE increases due to the increasing bias. Non-monotonic trend in MSE of $\hat{c}$ translates to a similar trend in the target generalization error under 0-1 loss.

2.2.2 Non-monotonicity in deep networks

The non-monotonic phenomenon is not an artifact of the linear model or the Gaussian assumptions. It appears across multiple datasets, architectures, and types of distribution shift. This section presents the key findings; full experimental details, including network architectures, training protocols, hyperparameter choices, and results for all target-OOD pairs, are given in Section A.5.

Geometric nuisances. On CIFAR-10, when the target task is a binary classification sub-task (e.g., Bird vs. Cat) and the OOD data consists of images from the same classes rotated by a fixed angle θ, the target generalization error is monotonically decreasing

for small θ but non-monotonic for larger angles (see the left panel of Fig. 2.4). A similar pattern appears with Gaussian blur: low blur yields monotonic improvement, high blur yields non-monotonicity. The transition occurs because small geometric perturbations are close enough to the target that they reduce variance without introducing substantial bias, while large perturbations are effectively a different distribution (see the middle panel of Fig. 2.4).

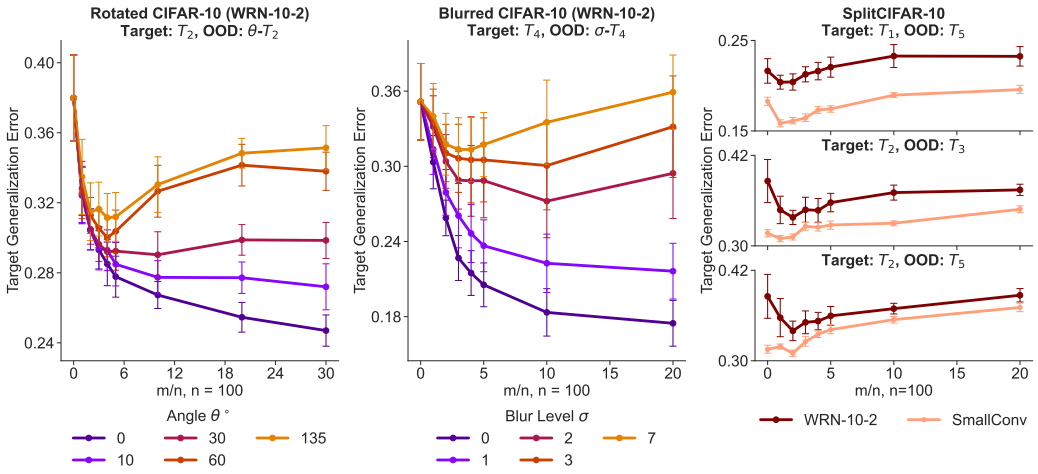


Figure 2.4: Left: Target: T_2 (Bird vs. Cat) from Split-CIFAR10; OOD: same images rotated by $\theta°$. Non-monotonic generalization curves appear for larger $\theta°$, with OOD samples actively hurting generalization (e.g., at $m/n = 20$ for 60° and 135°). **Middle:** Target: T_4 (Frog vs. Horse); OOD: images with varying Gaussian blur. Non-monotonic trends emerge at larger blur levels, while smaller blur levels show improvement with more OOD data. **Right:** Target generalization error vs. OOD sample count for 3 Split-CIFAR10 target-OOD pairs, across WRN and SmallConv architectures. All pairs exhibit non-monotonic trends under both models. Error bars indicate 95% confidence intervals (10 runs). See Section A.5 for details.

Semantic shifts. On five binary sub-tasks constructed from CIFAR-10 (Airplane vs. Automobile, Bird vs. Cat, Deer vs. Dog, Frog vs. Horse, Ship vs. Truck), using one sub-task as target and another as OOD produces non-monotonic trends across multiple target-OOD pairs and across two architectures (a small ConvNet and a wide

residual network). The non-monotonicity is not architecture-specific (see the right panel of Fig. 2.4).

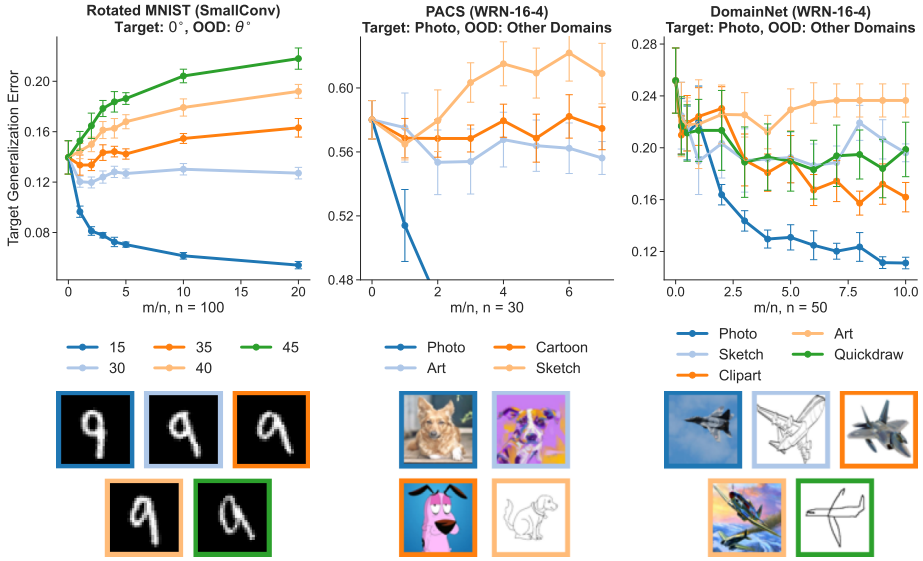


Figure 2.5: Non-monotonic trends in target generalization error on three DomainBed benchmarks. **Left:** Rotated MNIST (10 classes, 10 target samples/class, SmallConv), **Middle:** PACS (3 classes {dog, elephant, horse}, 10 target samples/class, WRN-16-4), and **Right:** DomainNet (2 classes {bird, plane}, 25 target samples/class, WRN-16-4). Error bars indicate 95% confidence intervals (10 runs).

Domain generalization benchmarks. On Rotated MNIST (DomainBed), PACS (Photo vs. Art/Cartoon/Sketch), and DomainNet (Photo vs. Sketch/Clipart/Quickdraw), non-monotonic trends in target generalization error appear for sufficiently dissimilar domains. On PACS, for instance, using sketches as OOD data for a photo-domain target produces a clear non-monotonic curve (see Fig. 2.5).

When non-monotonicity does not occur. On CINIC-10, a dataset combining CIFAR-10 with downsampled ImageNet images of the same classes, using ImageNet as OOD data produces a monotonically decreasing error. The shift between CIFAR-10

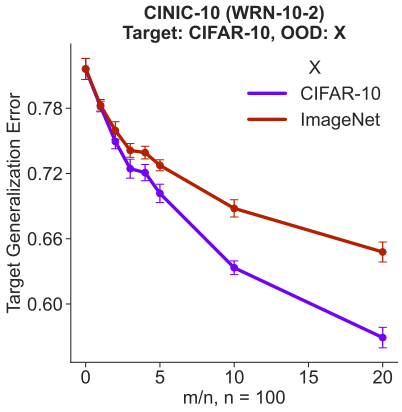


Figure 2.6: Target task is CIFAR-10 and OOD samples are from ImageNet. Although there is a distribution shift that causes the red curve to be higher error than the purple one, there is no non-monotonic trend in the generalization on CIFAR-10 due to OOD samples from ImageNet.

and ImageNet versions of the same class is small enough that the variance reduction from additional data always dominates the bias (see Fig. 2.6). Non-monotonicity requires the shift to be sufficiently large relative to the target sample size.

The effect of target sample size Non-monotonicity is most pronounced when the number of target samples n is small. As n increases, the target-only model approaches Bayes error and the marginal benefit of OOD data, whether positive or negative, shrinks. In the FLD example, non-monotonicity disappears for large n; in the deep-network experiments on MNIST and DomainNet, the same pattern is observed (see Fig. 2.7). However, as the Bayes-optimal error of the target task increases (for instance, by increasing the variance σ in the Gaussian mixture), non-monotonicity can persist even at larger n, because more samples are needed to approach Bayes error.

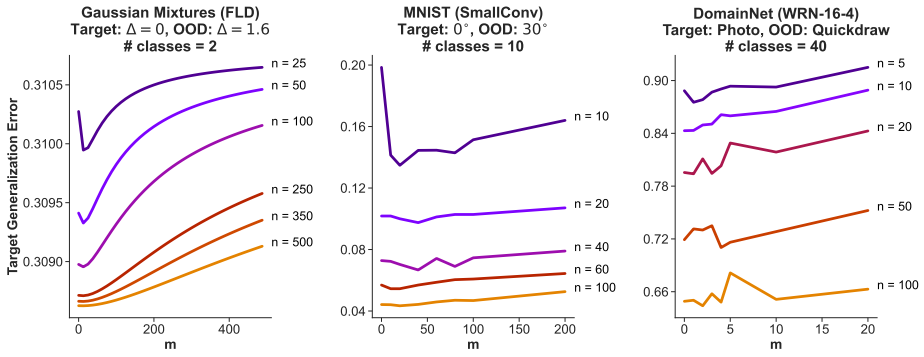


Figure 2.7: Target error (Y-axis) vs. OOD samples per class m (X-axis), for varying target samples per class n, across three target-OOD pairs: **left** Gaussian mixture ($\mu = 5, \sigma = 10$, $\Delta = 1.6$); **middle** 10-way rotated MNIST ($\theta = 30°$); **right** 40-way DomainNet (photo vs. quickdraw). Target error is computed analytically for the Gaussian case and empirically averaged over 10 and 3 random seeds for the other two. Non-monotonicity is observed at lower n; at larger n, OOD samples increase bias without reducing variance, yielding monotonically increasing error with m.

2.2.3 Standard remedies do not resolve non-monotonicity

Three natural strategies for mitigating the effect of OOD data were evaluated: data augmentation (random cropping and flipping), hyperparameter optimization (grid search over learning rate, weight decay, and architecture, using a target-only validation set), and pretraining followed by fine-tuning (pretrain on ImageNet, fine-tune on the combined target + OOD dataset). None of these strategies eliminate the non-monotonic trend. All three reduce the overall error level but preserve the qualitative shape: error decreases with small amounts of OOD data and increases with larger amounts (see Fig. 2.8). The non-monotonicity is robust to standard engineering interventions.

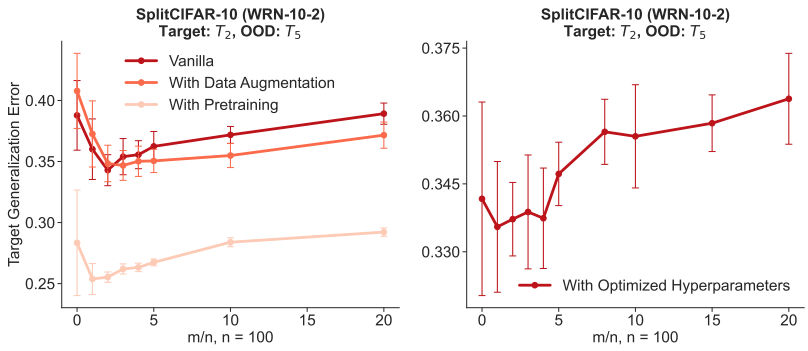


Figure 2.8: Left: For CIFAR-10 sub-task T_2 (Bird vs Cat) as target and T_5 (Ship vs Truck) as OOD, we train a WRN-10-2 with $n = 100$ target samples and varying OOD samples m under three settings: (1) Vanilla (darkest red), (2) Data augmentation (medium red), and (3) Pre-training on 14K CINIC-10 images followed by fine-tuning (lightest red). Despite reducing overall error, all three settings show deteriorating target generalization as the OOD fraction increases. **Right:** Hyper-parameter tuning via Ray over a target-only validation set still yields deteriorating generalization with more OOD samples. While such tuning is infeasible in practice (since sample identities are unknown), its failure here confirms the trend will persist in realistic settings. Error bars indicate 95% confidence intervals over 10 experiments.

2.2.4 Exploiting the non-monotonicity: weighted ERM in an oracle setting

The non-monotonicity documented above arises because the learner treats all samples, target and OOD, identically, minimizing the pooled empirical loss in Eq. (2.1). But if the learner knew which samples were OOD, it could weight them differently. The Ben-David, Blitzer, et al. (2010) bound suggests a natural approach: instead of minimizing the unweighted average, minimize an α-weighted combination of the target and OOD empirical losses,

$$\hat{h}_\alpha = \operatorname*{argmin}_h \left[\alpha \hat{e}_t(h) + (1-\alpha)\hat{e}_o(h)\right], \tag{2.3}$$

where $\hat{e}_t$ and $\hat{e}_o$ are the empirical losses computed separately on the n target and m OOD samples. The weight $\alpha \in [0,1]$ controls how much the learner trusts each distribution: $\alpha = 1$ ignores the OOD data entirely; $\alpha = 1/2$ weights them equally; intermediate values interpolate.

For the FLD example of Section 2.2.1, the generalization error of $\hat{h}_\alpha$ can be computed analytically for any α (see the derivation in Section A.4). The optimal weight α^* can then be found by numerical evaluation over $\alpha \in [0,1]$. The result is striking: regardless of m and regardless of the dissimilarity Δ between the two distributions, the generalization error under α^* is monotonically non-increasing in m (see Fig. 2.9). In other words, when the learner knows which samples are out-of-distribution and weights them optimally, every OOD sample helps, and the non-monotonicity disappears. The OOD data always reduces the variance of the hypothesis without introducing net bias,

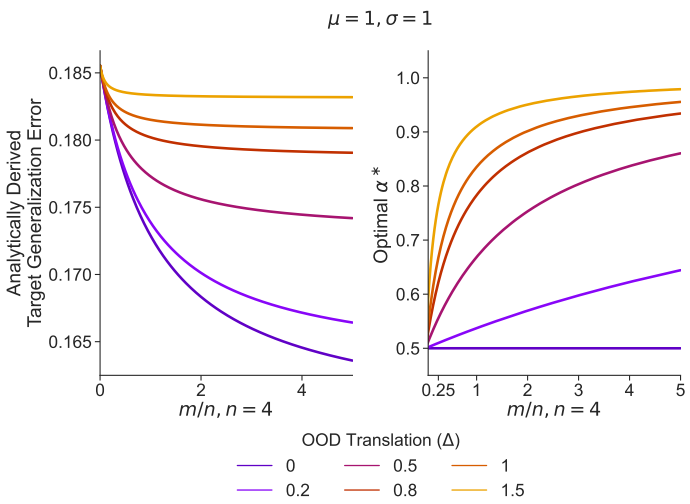


Figure 2.9: Left: Target generalization error using the weighted objective in Eq. (2.3) in FLD on the Gaussian mixture model. Unlike in Fig. 2.2, error decreases monotonically with OOD samples m. **Right:** Optimal weight α^* vs. m. α^* increases with m, more sharply for large Δ and gradually for small Δ; when $\Delta = 0$, $\alpha^* = 1/2$ for all m.

because α^* adjusts to downweight the OOD contribution as m grows and the bias term threatens to dominate.

The same phenomenon holds for deep networks. On several CIFAR-10 target-OOD pairs where the unweighted learner exhibits non-monotonic error, a learner that uses the α-weighted objective with α^* selected by grid search on a target-only validation set achieves monotonically decreasing error (see Fig. 2.10). Even a naive choice of $\alpha = 1/2$, which simply separates the target and OOD losses rather than pooling them, often rectifies the non-monotonicity, though the optimal α^* improves further.

These findings are encouraging but come with two critical caveats. First, the procedure requires knowing which samples are OOD — an assumption that is rarely satisfied in practice and that is itself a challenging detection problem. Second, it requires a

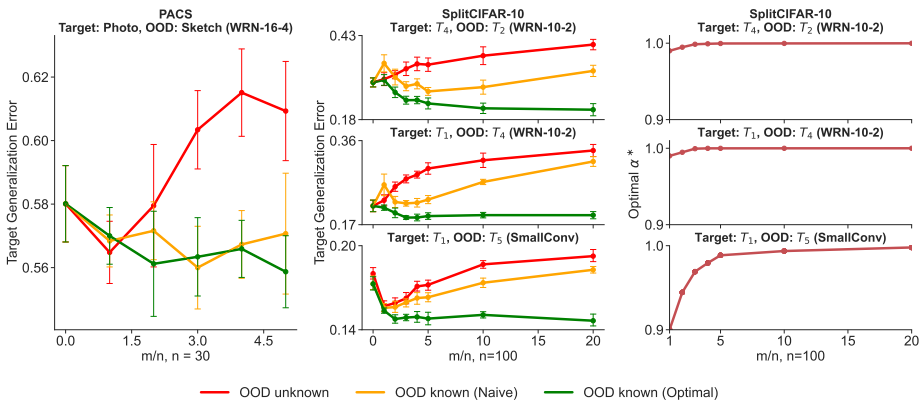


Figure 2.10: Three settings are compared: uniform average over all samples, agnostic to OOD identity ($\alpha = 1$, red); equal weighting of target and OOD average losses ($\alpha = 1/2$, yellow); and optimally weighted convex combination (green). The latter two require knowledge of sample identities. **Left/Middle:** Target generalization error vs. OOD sample count for PACS and CIFAR-10 sub-task pairs respectively. In PACS, $\alpha = 0.5$ yields a downward trend, likely due to greater target-OOD similarity, unlike in CIFAR-10. **Right:** Optimal α^* (via grid search) vs. OOD sample count for the CIFAR-10 pairs. α^* is close to but never exactly 1, indicating the weighted objective always provides some benefit. Error bars indicate 95% confidence intervals over 10 experiments.

held-out target validation set to tune α, which may not be available when target data is scarce. The weighted-ERM approach is therefore best understood as a proof of concept: it demonstrates that OOD data can always be made beneficial *in principle*, and that the non-monotonicity is an artifact of the agnostic treatment, not of the data itself. The practical challenge of exploiting OOD data without oracle knowledge of sample identities remains open.

The next section takes a different approach to the same problem. Instead of treating α as a hyperparameter to be tuned on validation data, it derives an approximately optimal α^* from a generative model that makes the relationship between source and target tasks explicit.

2.3 A tractable instance: approximately optimal domain adaptation with Fisher's Linear Discriminant

The preceding section established that non-monotonicity is a widespread phenomenon. This section asks a complementary question: *if we know which samples are OOD, can we exploit them optimally?* For a specific generative model — Gaussian class-conditionals with projection vectors drawn from a von Mises–Fisher (vMF) distribution — the answer is yes, and the optimal combination can be computed in closed form.

2.3.1 The generative model

Suppose each task $j \in \{0, 1, \ldots, J\}$ has a binary classification distribution

$$P^{(j)} = \pi^{(j)}\mathcal{N}(\nu^{(j)}, \Sigma^{(j)}) + (1 - \pi^{(j)})\mathcal{N}(-\nu^{(j)}, \Sigma^{(j)}),$$

where $\nu^{(j)}$ is the class-conditional mean (with the midpoint at the origin) and $\Sigma^{(j)}$ is the shared covariance. Task 0 is the target; tasks $1, \ldots, J$ are sources. For equal priors ($\pi = 1/2$) and shared covariance, the FLD projection vector $\omega^{(j)} = (\Sigma^{(j)})^{-1}\nu^{(j)}$ is a sufficient statistic for the classifier, and optimal classification reduces to thresholding at $\omega^{(j)\top}x = 0$.

The relationship between tasks is modeled by assuming the projection vectors are drawn from a common von Mises–Fisher distribution: $\omega^{(0)}, \omega^{(1)}, \ldots, \omega^{(J)} \overset{\text{iid}}{\sim} \mathcal{V}(\mu, \kappa)$, where μ is the mean direction and κ is the concentration parameter. When κ is large, the tasks are similar (projection vectors cluster tightly around μ); when κ is

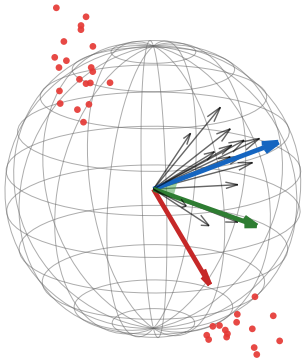


Figure 2.11: Geometric illustration of the generative assumptions, information constraints, and model class. Unit vectors shown: red dots/arrow = target data and projection estimate; black arrows = source projection vectors; blue arrow = average-source projection vector; green line = convex combinations of the red and blue arrows.

small, they are dissimilar. Fig. 2.11 illustrates the nature of the generative model in 2-dimensional space.

2.3.2 The class of classifiers

Define a class of classifiers parameterized by a convex combination of the target projection vector and the average-source projection vector:

$$\mathcal{H} = \left\{ h_\alpha(x) = \mathbb{1}\left\{ \left(\alpha\omega^{(0)} + (1-\alpha)\bar{\omega}\right)^\top x > 0 \right\} : \alpha \in [0,1] \right\},$$

where $\bar{\omega} = \frac{1}{J}\sum_{j=1}^{J} \omega^{(j)}$ is the average source projection vector. When $\alpha = 1$, the classifier uses only the target data; when $\alpha = 0$, it uses only the source data; intermediate values interpolate.

This parameterization induces a bias–variance tradeoff in task space, directly analogous

to the bias–variance tradeoff in threshold space observed in Section 2.2. For small n (few target samples), the target projection vector $\hat{\omega}^{(0)}$ is estimated with high variance, and an α closer to 0, favoring the lower-variance average-source vector, is preferred. For large n, the target estimate is precise and α closer to 1 is preferred. The optimal α^* trades off these two effects.

2.3.3 Approximating the optimal combination

The expected risk of a classifier h_α under the target distribution can be expressed in closed form. Given the projection vectors $\{\omega^{(j)}\}_{j=1}^{J}$ and the target class-conditional mean $\nu^{(0)}$ and covariance $\Sigma^{(0)}$, the risk under 0–1 loss is

$$\mathcal{E}(h_\alpha) = \mathbb{E}_{\omega_\alpha}\left[\Phi\left(\frac{-\omega_\alpha^\top \nu^{(0)}}{\sqrt{\omega_\alpha^\top \Sigma^{(0)} \omega_\alpha}}\right)\right],$$

where $\omega_\alpha = \alpha\omega^{(0)} + (1-\alpha)\mu$ is the convex combination of the target and average-source projection vectors and Φ is the CDF of the standard normal (see Section A.8 for the derivation). The expectation is over the distribution of ω_α, which inherits randomness from the estimation of both $\omega^{(0)}$ and μ. Despite the strong distributional assumptions, this expected risk is analytically intractable. The key idea is to approximate it using the asymptotic distributions of the estimated quantities.

The target projection vector $\hat{\omega}^{(0)}$ is asymptotically normal with mean $\omega^{(0)}$ and a covariance Σ_ω derived from the inverse-Wishart distribution of the estimated covariance matrix. The average-source direction $\hat{\mu}$ is asymptotically normal with mean μ and a

covariance Ψ derived from the vMF concentration. Since the two are independent, the convex combination $\hat{\omega}_\alpha = \alpha\hat{\omega}^{(0)} + (1-\alpha)\hat{\mu}$ is also asymptotically normal, with mean $\omega_\alpha = \alpha\omega^{(0)} + (1-\alpha)\mu$ and covariance $\Sigma_\alpha = \alpha^2\Sigma_\omega + (1-\alpha)^2\Psi$.

The approximation to the expected risk is then obtained by Monte Carlo: sample $\omega_\alpha^{(b)} \sim \mathcal{N}(\hat{\omega}_\alpha, \hat{\Sigma}_\alpha)$ for $b = 1, \ldots, B$ and compute

$$\hat{\mathcal{E}}(\alpha) = \frac{1}{B}\sum_{b=1}^{B} \Phi\left(\frac{-\omega_\alpha^{(b)\top}\hat{\nu}^{(0)}}{\sqrt{\omega_\alpha^{(b)\top}\hat{\Sigma}^{(0)}\omega_\alpha^{(b)}}}\right),$$

where $\hat{\nu}^{(0)}$ and $\hat{\Sigma}^{(0)}$ are plug-in estimates of the target class-conditional mean and covariance. The full procedure is:

1. Estimate $\hat{\omega}^{(0)}$, $\hat{\mu}$, $\hat{\Sigma}_\omega$, and $\hat{\Psi}$ from data.
2. For each α on a grid, draw B samples from $\mathcal{N}(\hat{\omega}_\alpha, \hat{\Sigma}_\alpha)$ and compute $\hat{\mathcal{E}}(\alpha)$.
3. Select $\alpha^* = \arg\min_\alpha \hat{\mathcal{E}}(\alpha)$.

2.3.4 Empirical validation

Simulations. In simulation, the approximated accuracy ($1 -$ risk) closely tracks the true (analytically intractable) accuracy for moderate to large n and J (see Fig. 2.12). The approximation is poorest for small n (where the asymptotic normal approximation to the target projection vector is inaccurate), but even then, the optimal classifier outperforms both the target-only and source-only classifiers.

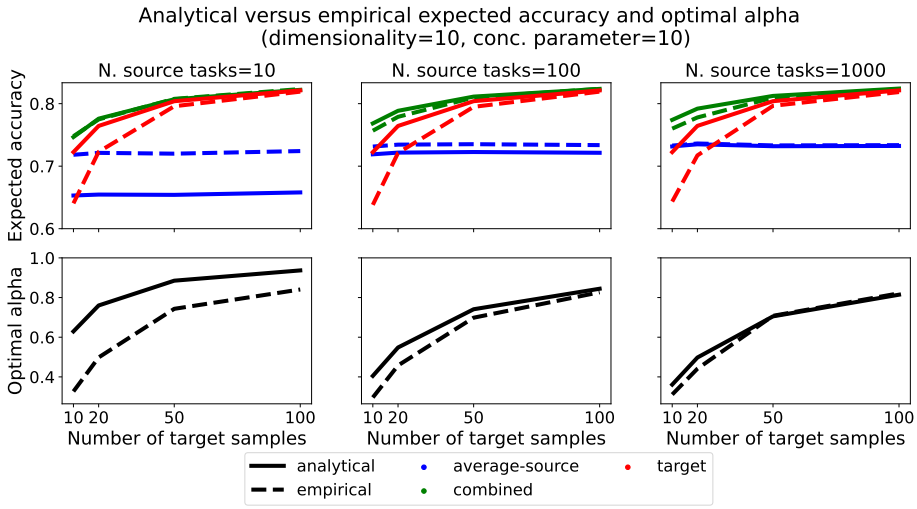


Figure 2.12: Validating our proposed approximation by comparing the approximated analytical accuracies and empirical accuracies and optimal convex coefficients α^* for different amounts of target training data n and number of source tasks J.

The effects of the generative parameters confirm the bias–variance interpretation. As κ increases (tasks become more similar), the optimal α^* decreases (favoring the source), because the source provides a good low-variance proxy. As the dimensionality d increases, the estimation error of the target projection vector grows, again favoring smaller α^*. As n increases, α^* increases toward 1, reflecting the diminishing need for the source regularization.

Physiological prediction. The method is evaluated on three physiological prediction tasks, namely EEG-based cognitive load classification (Section A.7.1), EEG-based stress classification (Section A.7.2), and ECG-based social stress classification (Section A.7.3), each involving multiple participants whose data constitute the source tasks and a target participant with limited data. In all three settings, the approximately optimal classifier outperforms or matches both the target-only and average-source

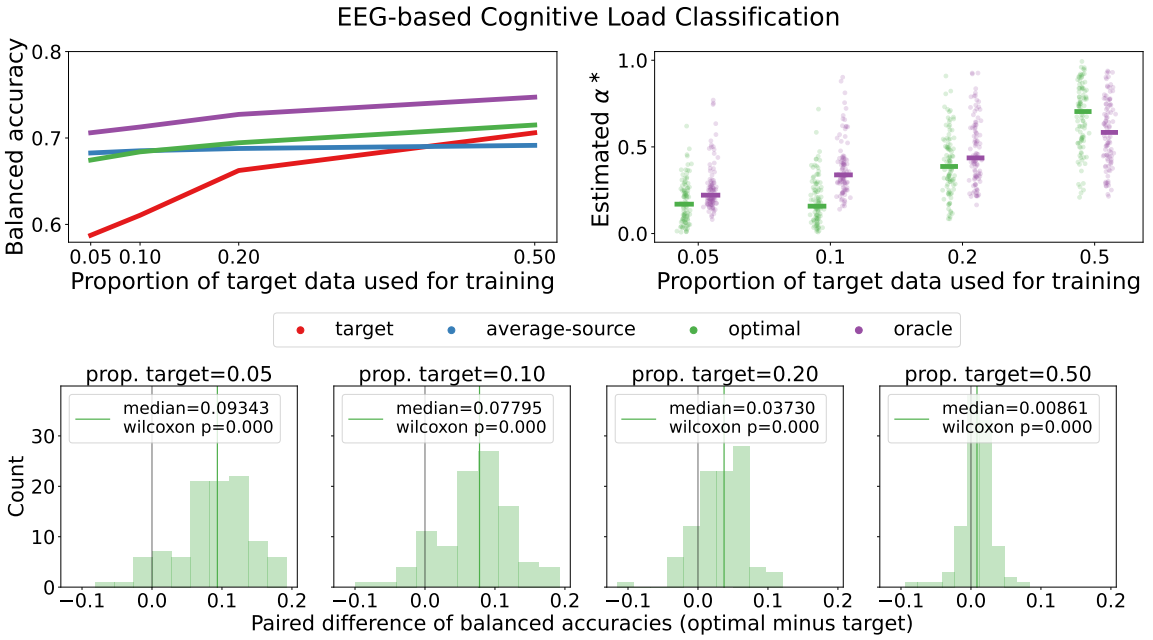


Figure 2.13: Balanced accuracy and relevant convex coefficients (**top**) and relative performance of the optimal and target classifiers (**bottom**) for the EEG-based cognitive load classification task.

classifiers, with the largest improvements occurring in the low-data regime. For the cognitive load task (see Fig. 2.13), the median balanced accuracy improvement over the target-only classifier is 9.3% when only 5% of the target data is used for training.

Privacy considerations An appealing property of the procedure is that it requires only the average source projection vector $\hat{\mu}$ and its standard error $\hat{\Psi}$, not the individual source projection vectors. These two summary statistics can be computed in the source environment and shared with the target device, preserving the privacy of individual source participants. This is relevant in physiological prediction settings where data-sharing agreements may prohibit transmitting raw data or individual-level classifiers.

2.3.5 Limitations

The analysis is restricted to the two-class problem with shared class-conditional covariance. Extension to the multi-class case is not straightforward, though one-vs-one reduction is a potential path. The assumption of a unimodal vMF distribution on the projection vectors may be violated in practice. Visualization of the projection vectors (see Helm and De Silva et al. (2024, Figure 7)) from the physiological datasets reveals multi-cluster structure, suggesting that a mixture of vMF distributions would be more appropriate. This would improve the approximation at the cost of additional parameters to estimate and a loss of the privacy-preserving property.

2.4 What the two-distribution framing cannot see

The results of this chapter share a common structure. A learner has access to data from two distributions, or, in Section 2.3, from a target and a collection of sources, and must combine them to make predictions about the target. The key design choice is the weight α assigned to the target samples relative to the non-current samples. When α is chosen well, optimally in the sense of the Ben-David, Blitzer, et al. (2010) bound or approximately in the sense of the FLD procedure of Section 2.3, the generalization error on the target improves.

This framing extends cleanly to a finite number of source distributions. With K sources, the learner chooses a vector of weights $(\alpha_1, \ldots, \alpha_K)$ on the simplex and minimizes the corresponding weighted empirical risk. The analysis proceeds by direct generalization of the two-distribution case.

But the framing does not survive the transition to infinitely many distributions. Suppose data arrives not from a fixed pair (P_t, P_o) or a fixed collection $\{P_1, \ldots, P_K\}$, but from an indefinite sequence $P_1, P_2, P_3, \ldots$ indexed by time. Three things go wrong simultaneously. First, the weight vector α becomes an infinite-dimensional object whose coordinates the learner cannot all estimate from finite data. Second, the notion of "target" disappears: every future moment is a new target, and no one of them is privileged. Third, and most importantly, the object the learner is asked to produce, a single hypothesis h that performs well across all distributions, may not exist.

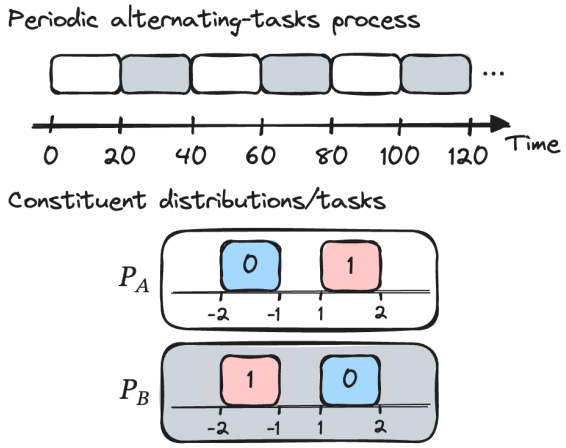


Figure 2.14: A process comprising of two classification distributions P_A and P_B over the same input space but with flipped class labels, arriving in alternating blocks. This is analogous to reversal learning in neuroscience, where an organism must detect and adapt to periodic reversals of stimulus-reward contingencies, except that the learner should anticipate the reversal rather than merely detect it after the fact.

The third failure is the decisive one. Consider two distributions P_A and P_B over the same input space but with flipped class labels, arriving in alternating blocks: P_A for the first 20 time steps, P_B for the next 20 time steps, P_A again for the following 20 time steps, and so on as illustrated in Fig. 2.14. The Bayes-optimal classifier for P_A is the worst possible classifier for P_B, and vice versa. No convex combination of the

two achieves better than chance on either. The weighted-ERM framework, no matter how carefully α is chosen, cannot solve this problem. The problem is not that α is hard to pick; the problem is that the output of the weighted-ERM procedure, a single hypothesis, is the wrong kind of object.

What is needed instead is a learner whose output is a *sequence* of hypotheses, or equivalently a hypothesis that takes time as an input. Such a learner can produce $h_t \approx h^*_{P_A}$ during the blocks where P_A is active and $h_t \approx h^*_{P_B}$ during the blocks where P_B is active, achieving near-Bayes risk at every moment. The next chapter develops the formalism for such learners, the notion of learnability that applies to them, and the main theorem of the thesis: that a suitably defined empirical risk minimizer over time-indexed hypotheses is a strong learner whenever the underlying stochastic process admits one.

Chapter 3

Prospective Learning

"The only reason for time is so that everything doesn't happen at once."

– attributed to Albert Einstein

3.1 From indexed families to stochastic processes

The preceding chapter ended with a concrete failure: when data arrives from a sequence of distributions rather than a fixed pair, the weighted-ERM framework cannot produce a hypothesis that performs well at every moment, because no such hypothesis need exist. What is required is a learner whose output is a sequence of hypotheses, one for each time, or equivalently a hypothesis that takes time as an input. Building the theory for such learners requires, first, a careful choice of what object to call "the data".

A natural first instinct is to model the data as an indexed family of distributions $\{P_t\}_{t\in\mathbb{N}}$, with P_t describing the distribution from which the learner's observation at time t is drawn. This was the framing adopted in earlier work on prospective learning (De Silva and Ramesh et al., 2023b), and it is sufficient for motivating the problem: the alternating-tasks example of Section 2.4 is expressed cleanly in this language, with $P_t = P_A$ for odd t and $P_t = P_B$ for even t. But as soon as one attempts to state a theorem, the indexed-family framing becomes inadequate, for three related reasons.

First, the framing cannot express dependence across time. If Z_t is drawn from P_t independently of Z_{t-1}, then the indexed family captures everything. But if Z_t depends on Z_{t-1}, as in a Markov chain, a hidden Markov model, or more general non-stationary processes, then a family of marginals is strictly less information than the joint distribution over all times. Many of the settings the thesis cares about, including reinforcement learning as it will be formalized in Chapter 4, exhibit exactly this kind of dependence.

Second, the framing does not support conditioning on realized history. A prospective learner, having observed the first t data points, must commit to a rule for predicting at all future times. The natural object in such an analysis is the learner's belief about the future given what it has seen, that is, the distribution of future observations conditioned on the realized past. Conditioning requires a joint probability space, not a family of marginals.

Third, and most fundamentally, the indexed-family framing does not give the learner's output a probabilistic status. The hypothesis selected by a prospective learner at time t is a function of the data $z_{\leq t}$, and as the data varies so does the hypothesis. Analyzing the learner's risk requires treating the hypothesis itself as a random variable, measurable with respect to the information available at time t. This in turn requires a filtration, which is not a concept an indexed family supports.

For these reasons, the thesis adopts the following setup. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. The data is a stochastic process

$$Z = (Z_t)_{t \in \mathbb{N}}$$

taking values in a measurable space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. Each $Z_t = (X_t, Y_t)$ is thus a pair of input and output random variables. The process Z generates a filtration $(\mathcal{F}_t)_{t \in \mathbb{N}}$ with $\mathcal{F}_t = \sigma(Z_{\leq t}) = \sigma(Z_1, \ldots, Z_t)$, the sigma-algebra generated by the first t observations.

Two notational conventions will be used throughout. The realized past is denoted by lower-case symbols, $z_{\leq t} = (z_1, \ldots, z_t) \in \mathcal{Z}^t$; the corresponding random variable is denoted by upper-case, $Z_{\leq t}$. This distinction matters because prospective risk, defined below, will be conditioned on the realized past, and one must be careful to keep the random and realized versions separate in the analysis. Similarly, the future from the learner's perspective at time t is denoted $Z_{>t} = (Z_{t+1}, Z_{t+2}, \ldots)$, with realizations $z_{>t}$.

The shift from $\{P_t\}$ to Z is not merely notational. It changes what questions the theory can ask. Indexed families permit questions of the form "how should the learner weight distributions $P_1, \ldots, P_t$," which is essentially the Chapter 2 question extended. Stochastic processes permit questions of the form "what can be inferred about the infinite future $Z_{>t}$ from a realization $z_{\leq t}$?," which is a question about prediction rather than weighting. The latter is the question of this thesis.

A final remark on the choice of $\mathbb{N}$ as the time index. The theory as developed here is in discrete time, which matches all of the empirical work in this and subsequent chapters. A continuous-time extension, in which Z is indexed by $\mathbb{R}_{\geq 0}$ and the learner observes a

sample path, is a natural direction but is not pursued here. The discrete-time setting is already rich enough to contain PAC learning as a special case, to exhibit the structural failures the theory is designed to address, and to admit the main learnability theorem of Section 3.4.

3.2 Hypothesis, loss, risk, and Bayes risk

With the data modeled as a stochastic process, the remaining pieces, namely what the learner outputs, what it is trying to minimize, and what the best achievable performance is, can be defined.

3.2.1 The hypothesis is a sequence

A classical learner outputs a single hypothesis $h : \mathcal{X} \to \mathcal{Y}$. A prospective learner cannot. The bridging argument of Section 2.4 showed that in settings like the alternating-tasks example, no single hypothesis is Bayes-optimal at every time, and so the object produced by the learner must be time-indexed. The thesis adopts the following definition:

Definition 3.2.1 (**Hypothesis sequence**)**.** A hypothesis sequence is an infinite sequence

$$h = (h_1, h_2, h_3, \ldots)$$

where each $h_t : \mathcal{X} \to \mathcal{Y}$. A *hypothesis class* $\mathcal{H} \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$ is a set of hypothesis sequences.

Two equivalent views of this object are useful. The first, just given, is as a sequence of ordinary hypotheses. The second is as a single function $h : \mathbb{N} \times \mathcal{X} \to \mathcal{Y}$ that takes

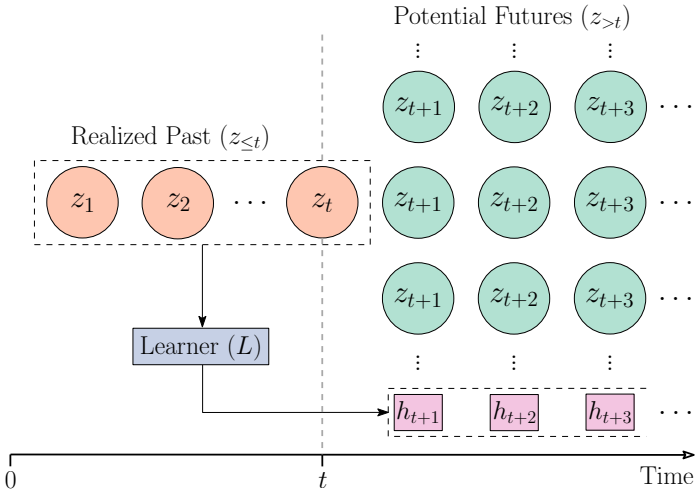


Figure 3.1: A schematic of the prospective learning framework.

both time and input as arguments. The equivalence $h(t, x) = h_t(x)$ is immediate. The second view will be convenient when discussing implementation in Sections 3.4 and 3.6, because it corresponds directly to how a prospective learner is built in practice: a standard neural network or decision tree that receives a time embedding alongside its usual input.

A particular sub-class is worth naming. A *time-agnostic* hypothesis sequence is one for which $h_t = h_{t'}$ for all $t, t' \in \mathbb{N}$, that is, the same predictor at every time. Time-agnostic hypotheses are the outputs of all the learners discussed in Chapter 2, and indeed of all classical PAC learners. The failure of this sub-class to contain Bayes-optimal sequences is, as will be shown in Section 3.3, a source of provable hardness.

3.2.2 Loss on the infinite future

The learner's performance at a single future moment s on a single datum (x_s, y_s) is measured by a per-step loss function $\ell : \mathbb{N} \times \mathcal{Y} \times \mathcal{Y} \to [0, 1]$, where $\ell(s, \hat{y}, y)$ is the

loss of predicting $\hat{y}$ when the true label is y, potentially depending on the time s. In most of what follows ℓ will not depend on the time s; for instance, the zero-one loss $\ell(s, \hat{y}, y) = \mathbb{1}\{\hat{y} \neq y\}$, but the time-dependent form is retained because some natural extensions of the framework require it.

A prospective learner is evaluated not on any single future moment but on the "entire" future. The primary loss the thesis considers aggregates the per-step losses over all times after t:

Definition 3.2.2 (**Prospective loss**)**.** For a hypothesis sequence h and a realization z of the stochastic process, the *prospective loss* at time t is

$$\bar{\ell}_t(h, z) = \limsup_{\tau \to \infty} \frac{1}{\tau} \sum_{s=t+1}^{t+\tau} \ell(s, h_s(x_s), y_s).$$

The limsup is the natural choice when ℓ is bounded, which guarantees the quantity is well-defined; when the Cesàro limit exists, it coincides with the lim. The division by τ gives the prospective loss the interpretation of a *time-averaged per-step loss over the infinite future*. A hypothesis sequence that achieves prospective loss 0.2 has, on average over the long run, a per-step error of 0.2.

Several variants are possible and some are used in later chapters. The most important is a discounted variant,

$$\bar{\ell}_t(h, z; \gamma) = (1 - \gamma) \sum_{s=t+1}^{\infty} \gamma^{s-t-1} \ell(s, h_s(x_s), y_s),$$

where $\gamma \in [0, 1)$ is a discount factor that weights more immediate future losses more heavily. The discounted form is useful when the stochastic process has an invariant distribution and the time-averaged loss is trivially equal to the invariant error, a situation that arises in Scenario 3 of Section 3.5 and motivates the connection to reinforcement learning in Chapter 4. For most of this chapter the time-averaged form suffices; the reader is directed to De Silva, Ramesh, and Yang et al. (2024, Appendix C) for the extensions to discounted losses.

3.2.3 Risk

The prospective loss $\bar{\ell}_t(h, z)$ is a function of the realization z, which is not directly accessible to the learner. The learner instead has access to $z_{\leq t}$ and must reason about the distribution of possible futures given what it has seen. This motivates:

Definition 3.2.3 (Prospective risk)**.** For a hypothesis sequence h measurable with respect to $\mathcal{F}_t$, the *prospective risk* at time t is

$$R_t(h) = \mathbb{E}\left[\bar{\ell}_t(h, Z) \,\middle|\, Z_{\leq t}\right].$$

Two features of this definition merit comment. First, the risk is a conditional expectation, as it depends on the realized past $Z_{\leq t}$, and in particular it is itself a random variable, measurable with respect to $\mathcal{F}_t$. This is a genuine departure from PAC learning, where the risk of a hypothesis is a deterministic function of the hypothesis and the (fixed, unconditioned) data distribution. In prospective learning, the "right answer" depends on what the learner has seen.

Second, the requirement that $h \in \mathcal{F}_t$ enforces that the hypothesis the learner produces at time t is a function of the data available at time t, ensuring the learner cannot peek into the future. This is implicit in any reasonable formalization but deserves stating explicitly.

3.2.4 Bayes risk

The benchmark against which a learner is measured is the best possible risk any $\mathcal{F}_t$-measurable hypothesis could achieve:

Definition 3.2.4 (Prospective Bayes risk)**.** The *prospective Bayes risk* at time t is

$$R_t^* = \inf_{h \in \mathcal{F}_t} R_t(h),$$

where the infimum is taken over all hypothesis sequences measurable with respect to $\mathcal{F}_t$.

Like $R_t(h)$, the Bayes risk R_t^* is a random variable depending on the realization $Z_{\leq t}$. In classical PAC learning, Bayes risk is a single number determined by the data distribution; here, it is a random variable because it depends on what the stochastic process has produced so far.

A useful sanity check. In the IID case where $Z_1, Z_2, \ldots$ are independent draws from a fixed distribution P, the Bayes risk R_t^* is almost surely equal to the classical Bayes risk of P and does not depend on t. The prospective framework recovers PAC learning as this special case. In the alternating-tasks example, R_t^* is zero whenever $\mathcal{H}$ contains the hypothesis sequence that alternates between the Bayes-optimal predictors for P_A

and P_B, and is bounded below by $1/2$ when $\mathcal{H}$ contains only time-agnostic sequences. The difference between these two Bayes risks, the one achievable with time-indexed hypotheses and the one achievable without, is the gap that time-agnostic ERM cannot close, and is the subject of Section 3.4.

3.2.5 Summary of Section 3.2

The setup is now complete. The data is a stochastic process Z on a probability space. The learner, having observed $z_{\leq t}$, produces a hypothesis sequence $h = (h_t)_{t\in\mathbb{N}}$, or equivalently a time-indexed function $h(t, x)$. Its performance is measured by the prospective risk $R_t(h)$, which is a conditional expectation of the time-averaged future loss given the realized past. The best achievable performance is R_t^*, the infimum of R_t over all hypothesis sequences measurable with respect to the past. In the remainder of the chapter, Section 3.3 shows that restricting to time-agnostic hypotheses is insufficient to achieve R_t^*; Section 3.4 gives a learner that does achieve it, under suitable conditions; Section 3.5 situates this result within a taxonomy of prospective learning problems; Sections 3.6 and 3.7 validate the theory empirically and generalize it beyond neural architectures.

3.3 Why time cannot be ignored: the failure of time-agnostic ERM

The definitions of Section 3.2 permit hypothesis sequences that vary arbitrarily across time. Classical learners do not produce such objects. An ERM learner trained on a dataset, whether drawn from one distribution or from many, outputs a single

hypothesis $h : \mathcal{X} \to \mathcal{Y}$ and deploys it identically at every future time. The question this section addresses is whether this restriction is harmless, that is, whether a learner that ignores time can still achieve prospective Bayes risk, or whether it constitutes a structural limitation.

The answer is that it constitutes a structural limitation. The section presents two constructions, each demonstrating a distinct mode of failure. In the first, time-agnostic ERM cannot even outperform a chance-level predictor. In the second, it can do better than chance but cannot approach Bayes risk. Together, these constructions show that time-agnostic ERM is not merely suboptimal but provably incapable, and that the gap is not one of sample complexity but one of representation.

3.3.1 Time-agnostic ERM

To make the claim precise, define the time-agnostic version of the prospective ERM problem. Let $\mathcal{G} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class of ordinary (non-time-indexed) predictors. The *time-agnostic hypothesis class* is the set of constant sequences built from $\mathcal{G}$:

$$\mathcal{H}_{\text{ta}} = \{h = (g, g, g, \ldots) : g \in \mathcal{G}\}.$$

Every element of $\mathcal{H}_{\text{ta}}$ satisfies $h_t = h_{t'}$ for all t, t'. A *time-agnostic ERM learner* is one that, given data $z_{\leq t}$, returns

$$\hat{h} = \arg\min_{h \in \mathcal{H}_{\text{ta}}} \frac{1}{t} \sum_{s=1}^{t} \ell(s, h_s(x_s), y_s).$$

Because $h_s = g$ for all s, this reduces to ordinary ERM over $\mathcal{G}$:

$$\hat{g} = \arg\min_{g \in \mathcal{G}} \frac{1}{t} \sum_{s=1}^{t} \ell(s, g(x_s), y_s),$$

with $\hat{h} = (\hat{g}, \hat{g}, \hat{g}, \ldots)$. This is the learner used by all classical PAC approaches, by the baselines in Chapter 2, and by existing continual and online learning methods that maintain a single set of parameters updated over time. The per-step loss may depend on s, but the hypothesis does not.

Example 1 (Time-agnostic ERM is not a weak prospective learner)**.** Consider a binary classification setting with $\mathcal{X} = \{-1, 1\}$, $\mathcal{Y} = \{0, 1\}$, and the zero-one loss $\ell(s, \hat{y}, y) = \mathbb{1}\{\hat{y} \neq y\}$. Define two distributions over $\mathcal{X} \times \mathcal{Y}$:

$$P_1(Y = 1 \mid X = x) = \begin{cases} \theta & \text{if } x = 1 \\ 1 - \theta & \text{if } x = -1 \end{cases}, \qquad P_2(Y = 1 \mid X = x) = \begin{cases} 1 - \theta & \text{if } x = 1 \\ \theta & \text{if } x = -1 \end{cases},$$

with $P_1(X = x) = P_2(X = x) = 1/2$ for both values of x, and $\theta \in (0, 1/2)$. The two distributions share the same marginal on $\mathcal{X}$ but have flipped class-conditional probabilities: the Bayes-optimal classifier for P_1 predicts $Y = 0$ when $X = 1$ and $Y = 1$ when $X = -1$, while for P_2 the predictions are reversed.

Now define the stochastic process Z by drawing $Z_t \sim P_1$ when t is odd and $Z_t \sim P_2$ when t is even, independently across time. This is the alternating-tasks process

encountered in Section 2.4.

Let $\mathcal{G}$ be any hypothesis class containing the Bayes-optimal classifiers for both P_1 and P_2. For any time-agnostic hypothesis $h = (g, g, \ldots)$, the prospective loss decomposes as

$$\bar{\ell}_t(h, Z) = \lim_{\tau \to \infty} \frac{1}{2\tau} \sum_{s=t+1}^{t+2\tau} \ell(s, g(X_s), Y_s) = \frac{1}{2} \left[R_{P_1}(g) + R_{P_2}(g) \right]$$

almost surely, where $R_{P_i}(g) = \mathbb{E}_{(X,Y)\sim P_i}[\mathbb{1}\{g(X) \neq Y\}]$ is the classical risk of g under distribution P_i. The key observation is that for any classifier g,

$$R_{P_1}(g) + R_{P_2}(g) = 1.$$

To see this, fix any $x \in \{-1, 1\}$ and any prediction $\hat{y} = g(x)$. If $g(x)$ agrees with P_1's Bayes-optimal prediction at x, then it disagrees with P_2's (since their class-conditional probabilities are complementary), and vice versa. The expected loss at x under one distribution plus the expected loss at x under the other is exactly 1. Averaging over x gives $R_{P_1}(g) + R_{P_2}(g) = 1$.

It follows that $\bar{\ell}_t(h, Z) = 1/2$ almost surely, for every time-agnostic hypothesis h, and therefore $R_t(h) = 1/2$ for all t and all $h \in \mathcal{H}_{\text{ta}}$. A learner that predicts by fair coin flip, achieving chance-level performance, also achieves prospective risk $1/2$. The time-agnostic learner cannot outperform chance.

Meanwhile, the Bayes risk R_t^* over the full (time-indexed) hypothesis class is zero: the hypothesis sequence h^* with h_t^* = Bayes-optimal classifier for P_1 at odd t and Bayes-optimal for P_2 at even t achieves zero loss at every time step.

The gap between $R_t^* = 0$ and $\inf_{h \in \mathcal{H}_{\text{ta}}} R_t(h) = 1/2$ is as large as it can be. Time-agnostic ERM is not merely a poor prospective learner; it is not a prospective learner at all.

The first example might seem pathological, as the class labels are maximally adversarial between the two distributions. One might hope that in less extreme settings, time-agnostic ERM at least outperforms chance. The second example shows that this can happen, yet even then, time-agnostic ERM cannot approach Bayes risk.

Example 2 (Time-agnostic ERM is weak but not strong)**.** Modify the setup so that $\mathcal{X} = \{-1, 0, 1\}$, with $P_1(X = x) = P_2(X = x) = 1/3$ for all x. The class-conditional probabilities are:

$$P_1(Y = 1 \mid X = x) = \begin{cases} \theta & \text{if } x \leq 0 \\ 1 - \theta & \text{if } x = 1 \end{cases}, \qquad P_2(Y = 1 \mid X = x) = \begin{cases} 1 - \theta & \text{if } x \geq 0 \\ \theta & \text{if } x = -1 \end{cases}.$$

The distributions agree at $x = -1$ (both have $P(Y = 1 \mid X = -1) \in \{\theta, 1 - \theta\}$ with opposite values) and at $x = 1$ (similarly), but they disagree at $x = 0$: under P_1, the Bayes-optimal prediction at $x = 0$ is $Y = 0$; under P_2 it is $Y = 1$. No single classifier can be Bayes-optimal at $x = 0$ under both distributions simultaneously.

As before, let $Z_t \sim P_1$ for odd t and $Z_t \sim P_2$ for even t. For any time-agnostic hypothesis $h = (g, g, \ldots)$, the prospective loss is $\frac{1}{2}[R_{P_1}(g)+R_{P_2}(g)]$. A short calculation shows that any classifier g must incur a combined loss of at least $1/3$: it can be correct at $x = -1$ and $x = 1$ under both distributions (since the labels at these points are the same across distributions, which constitutes the "agreement" region), but at $x = 0$ it incurs an error of at least $\min(\theta, 1 - \theta)$ under one distribution. The minimum over $\mathcal{H}_{\text{ta}}$ of the prospective risk is therefore bounded below by $1/3$.

A chance-level predictor (predict $\hat{y} \in \{0, 1\}$ uniformly at random) achieves prospective risk $1/2$. Given enough data, ERM over $\mathcal{G}$ will learn the correct labels at $x = -1$ and $x = 1$ — these are shared across both distributions — bringing the risk below $1/2$. Specifically, with t samples, the probability that ERM has not seen at least one observation at $x = 1$ and one at $x = -1$ is at most $2 \cdot 3^{-t}$, and once it has, it achieves risk at most $1/3 + \epsilon$ for $\epsilon = O(t^{-1/2})$.

So time-agnostic ERM beats chance, making it a weak prospective learner for this process. But R_t^* over the full time-indexed hypothesis class is again zero, and $\inf_{h \in \mathcal{H}_{\text{ta}}} R_t(h) \geq 1/3 > 0$ for all t. The learner is not a strong prospective learner.

The two examples together prove the following:

Proposition 3.3.1. There exist stochastic processes for which time-agnostic ERM is not a weak prospective learner. There also exist stochastic processes for which time-agnostic ERM is a weak prospective learner but not a strong one.

The failures in Examples 1 and 2 are not failures of approximation, optimization, or sample size. They are failures of *representation*: the hypothesis class $\mathcal{H}_{\text{ta}}$ does not

contain the right object. No amount of data, no choice of optimization algorithm, and no regularization strategy can close the gap between $\inf_{h \in \mathcal{H}_{\text{ta}}} R_t(h)$ and R_t^*, because the gap is between the best constant sequence and the best time-varying sequence.

This observation has a direct practical consequence. Online learning methods such as Follow-the-Leader (Nicolo Cesa-Bianchi and Lugosi, 2006), online SGD, and Bayesian gradient descent (Zeno et al., 2021), all of which maintain and update a single set of parameters, produce time-agnostic hypotheses. They may track a non-stationary distribution in the sense that their parameters change as new data arrives, but at any given moment their output is a single predictor deployed identically across all future times. Proposition 3.3.1 says that this is structurally insufficient. The learner must be able to produce different predictions at different future times, which requires either maintaining an explicit sequence of predictors or, equivalently, taking time as an input.

Whether time-agnostic weak and strong learnability are equivalent in the distribution-agnostic setting remains an open question. In PAC learning, the equivalence of weak and strong learnability is a celebrated theorem; here, the two constructions show that no analogous equivalence can hold for all stochastic processes unless the hypothesis class is allowed to be time-indexed. This question is revisited in Chapter 5.

The next section introduces a learner that does take time as input and proves that it achieves strong learnability.

3.4 Prospective ERM and the main learnability result

The preceding section established that time-agnostic ERM is structurally incapable of achieving Bayes risk for certain stochastic processes. The gap is not one of sample complexity but of representation: the hypothesis class does not contain the right object. This section closes the gap. It defines two notions of prospective learnability, strong and weak, introduces a learner called prospective ERM that operates over time-indexed hypothesis classes, and proves that prospective ERM is a strong prospective learner under two conditions: a consistency condition analogous to the one in PAC learning, and a uniform-concentration condition on the empirical estimate of the prospective loss.

3.4.1 Learnability

Two levels of prospective learnability are defined, paralleling the classical distinction in PAC learning.

Definition 3.4.1 (Strong prospective learnability)**.** A family $\mathcal{Z}$ of stochastic processes is *strongly prospectively learnable* if there exists a learner L with the following property: for all $\epsilon, \delta > 0$ and for any stochastic process $Z \in \mathcal{Z}$, there exists a time $t'(\epsilon, \delta)$ such that for any $t > t'$, the learner L, given data $z_{\leq t}$, outputs a hypothesis sequence $\hat{h} \in \mathcal{F}_t$ satisfying

$$\mathbb{P}\left[R_t(\hat{h}) - R_t^* < \epsilon\right] \geq 1 - \delta.$$

In words: after observing enough data, the learner's prospective risk is arbitrarily close to Bayes risk with arbitrarily high probability, uniformly over the family of processes.

The definition mirrors PAC learnability with one crucial difference. In PAC learning, Bayes risk R^* is a constant determined by the data distribution; here, R_t^* depends on the realized past $z_{\leq t}$ and is itself a random variable. The learner must match a moving target — not because the target drifts adversarially, but because what constitutes optimal behavior depends on what the stochastic process has revealed so far.

Not all families of stochastic processes admit strong learnability. Proposition 1 exhibited a process for which time-agnostic ERM can beat chance but not achieve Bayes risk. This motivates a weaker notion.

Definition 3.4.2. Weak prospective learnability A family $\mathcal{Z}$ of stochastic processes is *weakly prospectively learnable* if there exists a learner L and an $\epsilon > 0$ with the following property: for any $\delta > 0$, there exists a time $t'(\epsilon, \delta)$ such that for any stochastic process $Z \in \mathcal{Z}$ and any $t > t'$,

$$\mathbb{P}\left[R_t^0 - R_t(\hat{h}) > \epsilon\right] \geq 1 - \delta,$$

where R_t^0 is the prospective risk of a chance-level predictor, one that predicts $\mathbb{E}[Y]$ at every time.

A weakly prospectively learnable family is one for which some learner consistently beats chance. The two examples of Section 3.3 showed that time-agnostic ERM can fail to be even a weak learner (Example 1), or can be weak but not strong (Example 2). Whether weak and strong prospective learnability are equivalent in general remains an open question, in contrast to PAC learning where their equivalence is a celebrated result. The relationship between the two notions is revisited in Chapter 5.

3.4.2 The prospective ERM learner

In PAC learning, ERM selects the hypothesis that minimizes the empirical loss on training data, and the fundamental theorem of statistical learning says that ERM is a strong learner when the hypothesis class has finite VC dimension. The prospective analog of this story begins with an analog of ERM.

The key idea is simple: instead of minimizing the empirical loss using a time-agnostic hypothesis, minimize an empirical estimate of the prospective loss using a time-indexed hypothesis. The complication is that the prospective loss involves a limsup over the infinite future, which the learner cannot observe. The empirical analog replaces this limsup with a maximum over partial averages computed from the training data.

Formally, given data $z_{\leq t}$, define the empirical partial average of hypothesis h computed from the first m samples as

$$e_m(h) = \frac{1}{m} \sum_{s=1}^{m} \ell(s, h_s(x_s), y_s).$$

The prospective loss $\bar{\ell}_t(h, Z)$ is the limsup of $e_m(h)$ as $m \to \infty$, but the learner has access only to $e_m(h)$ for $m \leq t$. The empirical surrogate for the limsup is

$$\max_{u \leq m \leq t} e_m(h),$$

where u is a lower bound on the window over which the maximum is taken. The

role of u is to discard early samples, where the empirical average is a poor estimate; its precise value will depend on the convergence rate of the concentration condition below.

Definition 3.4.3 (Prospective ERM)**.** Let $\{\mathcal{H}_t\}_{t=1}^{\infty}$ be an increasing sequence of hypothesis classes with each $\mathcal{H}_t \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$. Let $\{i_t\}_{t=1}^{\infty}$ be an increasing sequence of indices with $i_t \leq t$, and let $\{u_{i_t}\}$ be a corresponding sequence with $u_{i_t} \leq i_t$ and $u_{i_t} \to \infty$. A *prospective ERM learner* is one that, given data $z_{\leq t}$, returns

$$\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}_{i_t}} \max_{u_{i_t} \leq m \leq t} \frac{1}{m} \sum_{s=1}^{m} \ell(s, h_s(x_s), y_s).$$

Several features of this definition deserve comment.

First, the hypothesis class $\mathcal{H}_{i_t}$ grows with time. This is a standard device in learning theory: at early times when data is scarce, the learner should use a small hypothesis class to avoid overfitting; as data accumulates, the class expands. The precise rate at which $\mathcal{H}_{i_t}$ grows is determined by the concentration condition in Theorem 1 below.

Second, the inner maximum over m is the empirical analog of the limsup. It ensures that the learner selects a hypothesis that performs well not just on average over the training data but on the worst partial average over a range of window sizes. This is stronger than ordinary ERM and reflects the requirement that the learner's performance must hold up over the infinite future, not merely over a fixed test set.

Third, the sequence $\{i_t\}$ depends only on the concentration rate γ_t of condition (ii) below, not on the data. This is a technical requirement that ensures the proof goes

through; in practice, one can set $i_t = t$ and $u_{i_t} = t$, which reduces the definition to the simpler form

$$\hat{h} = \arg\min_{h \in \mathcal{H}_t} \frac{1}{t} \sum_{s=1}^{t} \ell(s, h_s(x_s), y_s).$$

This simpler form changes only the sample complexity, not the asymptotic guarantee.

3.4.3 The main theorem

The central result of the thesis is the following.

Theorem 3.4.1 (Prospective ERM is a strong prospective learner)**.** Consider a finite family $\mathcal{Z}$ of stochastic processes. Suppose the following two conditions hold.

(i) Consistency. There exists an increasing sequence of hypothesis classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ with each $\mathcal{H}_t \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$, such that for all $Z \in \mathcal{Z}$,

$$\lim_{t \to \infty} \mathbb{E}\left[\inf_{h \in \mathcal{H}_t} R_t(h) - R_t^*\right] = 0, \tag{3.1}$$

where the infimum is over hypothesis sequences h measurable with respect to $\mathcal{F}_t$.

(ii) Uniform concentration of the limsup. For all $Z \in \mathcal{Z}$,

$$\mathbb{E}\left[\max_{h \in \mathcal{H}_t} \left| \bar{\ell}_t(h, Z) - \max_{u_t \le m \le t} e_m(h) \right| \right] \le \gamma_t, \tag{3.2}$$

for some sequence $\gamma_t \to 0$ and some sequence $u_t \to \infty$ with $u_t \le t$, both uniform over $\mathcal{Z}$.

Then there exists a sequence $\{i_t\}$ depending only on γ_t such that the prospective ERM learner that returns $\hat{h}$ as defined above is a strong prospective learner for the family $\mathcal{Z}$.

The two conditions are direct analogs of the two pillars of PAC learning theory.

Condition (i), consistency, says that the expanding hypothesis classes eventually contain a hypothesis sequence whose risk is close to Bayes risk. This is the analog of the requirement in PAC learning that the hypothesis class contain a good approximation to the Bayes classifier. The expectation is over the realized past; because R_t^* is a random variable, mere pointwise approximation would not suffice.

Condition (ii), uniform concentration, says that the limsup of the per-step loss (the prospective loss) is well-approximated by the maximum partial average computed from training data, uniformly over the hypothesis class. This is the analog of uniform convergence in PAC learning, which requires that empirical risk converge to population risk uniformly over the hypothesis class. The key difference is that in PAC learning the population risk is a fixed integral with respect to a fixed distribution, whereas here the prospective loss is itself a limsup (a limit of an infinite sequence) and the empirical surrogate is a maximum over finitely many partial averages. The rate γ_t at which this approximation improves determines the sample complexity.

Proof. The proof proceeds in three steps. Full details are given in Section B.1.

1. *Existence of a good reference hypothesis*: By condition (i), there exists a sequence of $\mathcal{F}_t$-measurable hypothesis sequences $h^{(t)} \in \mathcal{H}_t$ whose expected prospective

risk converges to Bayes risk. A Borel-Cantelli argument along a carefully chosen subsequence converts this convergence in expectation to almost-sure convergence of the empirical surrogate: there exists a (random) time t_0 after which

$$\mathbb{E}\left[\sup_{u_t \le m \le \infty} e_m(h^{(t)}) \;\middle|\; Z_{\le t}\right] - R_t^* \to 0$$

almost surely. The subsequence construction follows Hanneke (2021) and ensures that the indices i_t depend only on γ_t, not on the data.

2. *Prospective ERM does at least as well*: By definition, $\hat{h}$ minimizes $\max_{u_{i_t} \le m \le t} e_m(h)$ over $\mathcal{H}_{i_t}$. Since $h^{(t)} \in \mathcal{H}_{i_t}$ (because $i_t \le t$ and $\mathcal{H}_{i_t} \subseteq \mathcal{H}_t$), the empirical surrogate of $\hat{h}$ is at most that of $h^{(t)}$.

3. *Connecting empirical surrogate to prospective loss*: Condition (ii), combined with a Borel-Cantelli argument over the subsequence $\{i_t\}$, ensures that the empirical surrogate and the true prospective loss are close for all hypotheses in $\mathcal{H}_{i_t}$ simultaneously. Combining Steps 2 and 3 gives

$$\bar{\ell}_t(\hat{h}, Z) \le \sup_{u_{i_t} \le m \le \infty} e_m(h^{(t)}) + \sqrt{\gamma_{i_t}},$$

and taking conditional expectations and using Step 1 yields $R_t(\hat{h}) - R_t^* \to 0$ almost surely. The bounded convergence theorem converts this to convergence in probability, completing the proof.

□

3.4.4 Sufficient conditions: a countable hypothesis class suffices

The conditions of Theorem 3.4.1 are stated abstractly. The natural follow-up question is: for what kinds of hypothesis classes and stochastic processes are they satisfied? The following result gives a concrete sufficient condition.

Theorem 3.4.2. Consider a finite family $\mathcal{Z}$ of stochastic processes. If there exists a countable hypothesis class $\mathcal{H} \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$ such that for all $Z \in \mathcal{Z}$,

$$\lim_{t\to\infty} \mathbb{E}\left[\inf_{h\in\mathcal{H}} R_t(h) - R_t^*\right] = 0,$$

where $h \in \mathcal{H}$ is measurable with respect to $\mathcal{F}_t$, then there exist $\mathcal{H}_t$, u_t, and γ_t satisfying the two conditions of Theorem 3.4.1.

Proof. The construction follows Hanneke (2021) Section 4. For a finite hypothesis class, the sequence u_t is built by tracking, for each $h \in \mathcal{H}$, the earliest time t_u^h after which the maximum partial average is within 2^{-u} of the limsup. Taking the minimum over h and choosing u via a quantile argument over realizations of Z gives a sequence $u_t \to \infty$ such that condition (ii) holds. The construction extends to a countable $\mathcal{H}$ by enumerating its elements and defining $\mathcal{H}_t$ as the first t elements; the rate γ_t is then controlled by the enumeration. Full details are in Section B.2. □

In PAC learning, one first proves uniform convergence for a finite hypothesis class, then extends to infinite classes via VC dimension and covering numbers. Theorem 3.4.2 should be understood in the same spirit: it is a first step toward characterizing the sample complexity of prospective learning, not the final word. Extensions to infinite

hypothesis classes, and the identification of a complexity measure analogous to VC dimension for sequences of hypotheses, remain open.

3.4.5 The periodic case: explicit sample complexity

For one class of stochastic processes, periodic processes, the sample complexity of prospective ERM can be computed exactly. Suppose the stochastic process has period T: $Z_t \sim P_{(t \bmod T)}$ for distributions $P_0, \ldots, P_{T-1}$, with independence across time. Let $\mathcal{G}$ be a hypothesis class with finite VC dimension that contains the Bayes-optimal classifier for each P_k. Define

$$\mathcal{H}_T = \{h : h_{t+T} = h_t \text{ and } h_t \in \mathcal{G} \text{ for all } t\}.$$

This is the class of T-periodic hypothesis sequences built from $\mathcal{G}$. Prospective ERM over $\mathcal{H}_T$ reduces to T independent ERM problems, one for each phase of the period, each using the approximately t/T samples that fall in that phase. A covering-number argument shows that if

$$t \geq \max\left\{\frac{64}{\epsilon^2}\log\frac{4\,C(\epsilon, \mathcal{G}^T)}{\delta},\ \frac{16T}{\epsilon^2}\right\},$$

where $C(\epsilon, \mathcal{G}^T)$ is the ϵ-covering number of the class of T-length sequences from $\mathcal{G}$, then $\mathbb{E}[R_t(\hat{h})]$ is within 2ϵ of the limiting Bayes risk with probability at least $1 - \delta$. The sample complexity is dominated by the first term and grows at most linearly in T, as one would expect: learning a periodic process of period T requires seeing at

least one full period.

This calculation also illuminates what happens when the period T is unknown. The learner can use the union $\bigcup_{T\in\mathbb{N}} \mathcal{H}_T$ as its hypothesis class; this is countable, so Theorem 3.4.2 applies. Prospective ERM is a strong learner for any periodic process with bounded period, even without knowing the period in advance.

3.4.6 Discounted losses

Some stochastic processes have invariant distributions, in which case the time-averaged prospective loss $\bar{\ell}_t$ can be trivially equal to the invariant risk for any hypothesis, making prospective learning vacuous. The natural remedy is a discounted loss, as introduced in Section 3.2. The learnability results extend:

Corollary 3.4.1 (Discounted losses)**.** If the conditions of Theorem 3.4.1 hold and there exists a constant $c > 0$ such that for all $Z \in \mathcal{Z}$, all $t \in \mathbb{N}$, and all $h \in \mathcal{H}_t$,

$$R_t(h;\mu) - R_t^*(\mu) \leq c\left[R_t(h;1/\tau) - R_t^*(1/\tau)\right],$$

where $R_t(\cdot;\mu)$ denotes the discounted risk and $R_t(\cdot;1/\tau)$ the time-averaged risk, then prospective ERM (implemented with the time-averaged loss) is also a strong learner for the discounted risk.

The condition asks that the gap in discounted risk is uniformly dominated by the gap in time-averaged risk. When this holds, convergence of the time-averaged gap to zero, established by Theorem 3.4.1, implies convergence of the discounted gap as well. The proof is a one-line application of Markov's inequality.

3.4.7 How to implement prospective ERM

The definitions above involve hypothesis classes of infinite sequences and a minimax optimization. In practice, prospective ERM is much simpler.

Suppose the base hypothesis class consists of neural networks, such as multi-layer perceptrons for tabular data or convolutional networks for images. A time-agnostic learner trains such a network on pairs (x_s, y_s) to minimize cross-entropy. A prospective learner does the same, with one modification (see Fig. 3.2): the network receives a time embedding $\varphi(s)$ as an additional input alongside x_s, and is trained on triples $(\varphi(s), x_s, y_s)$.

At training time, the network sees data $\{(\varphi(s), x_s, y_s)\}_{s=1}^{t}$ and minimizes the empirical loss

$$\frac{1}{t}\sum_{s=1}^{t} \ell(s, f_\theta(\varphi(s), x_s), y_s),$$

where f_θ is the parameterized network. At inference time, to predict at a future time $t' > t$, the network receives the input $(\varphi(t'), x_{t'})$ and outputs a prediction $\hat{y}_{t'} = f_\theta(\varphi(t'), x_{t'})$. The hypothesis sequence is never explicitly materialized; it is implicitly encoded in the network's dependence on the time embedding.

The time embedding $\varphi : \mathbb{N} \to \mathbb{R}^d$ must be chosen to match the temporal structure of the problem. A Fourier embedding,

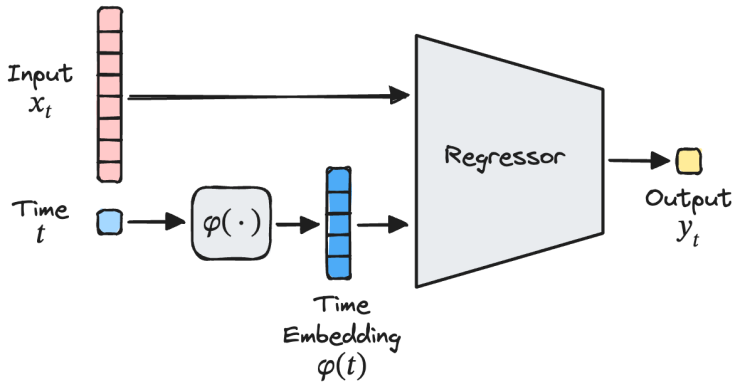


Figure 3.2: A schematic of a prospective learner that receives time as an additional input alongside the input.

$$\varphi_f(t) = \left(\sin(\omega_1 t), \ldots, \sin(\omega_{d/2} t), \cos(\omega_1 t), \ldots, \cos(\omega_{d/2} t)\right),$$

with frequencies $\omega_i = \pi/i$ for $i = 1, \ldots, d/2$, is effective for periodic processes. A monomial embedding,

$$\varphi_m(t) = \left(t, t^2, \ldots, t^d\right),$$

is effective for processes whose distribution drifts polynomially. The Fourier embedding shares some similarity with the positional encoding used in the transformer architecture (Vaswani et al., 2017). As Section 3.6 will show, the choice of embedding can be the difference between converging to Bayes risk and failing entirely, an observation with both practical and theoretical significance.

The gap between the theory (infinite sequences, countable hypothesis classes, limsup

convergence) and the practice (a neural network with a time embedding, trained by SGD) is worth acknowledging explicitly. Theorem 3.4.1 guarantees that prospective ERM achieves strong learnability; it does not guarantee that SGD on a finite-width network finds the ERM solution. The experiments of Section 3.6 provide empirical evidence that the gap is small in practice, but closing it theoretically, by providing approximation and optimization guarantees for specific architectures, for example, remains an important open problem.

3.4.8 Summary

The main result of this section, and of the thesis, is that prospective ERM, which is ERM over time-indexed hypotheses, is a strong prospective learner under conditions analogous to those that make classical ERM consistent in PAC learning. The learner's key property is that it makes time an input. This is what Section 3.3 showed was necessary; Theorem 3.4.1 shows it is sufficient. The periodic case gives explicit sample complexity; Theorem 3.4.2 shows that a countable hypothesis class is enough; and the implementation remark reduces the entire construction to a single practical operation: feed a time embedding to a standard network.

3.5 A taxonomy of prospective learning problems

The formalism of Sections 3.1 to 3.4 is stated in full generality: a stochastic process Z, a hypothesis sequence h, a prospective risk $R_t(h)$. But stochastic processes vary enormously in their dependence structure, and so does the difficulty of prospective learning. This section organizes the landscape into four scenarios, ordered by the

richness of the temporal structure the learner must contend with. Each scenario recovers or extends a known learning paradigm, making precise the sense in which prospective learning unifies existing frameworks as special cases. A fifth setting, covering processes with infinitely many novel tasks, is discussed separately, as it illustrates a phenomenon not captured by any of the four canonical scenarios.

3.5.1 Scenario 1: Independent and identically distributed data

This is the simplest case. The stochastic process Z satisfies $P_{Z_{t'}|Z_{\leq t}} = P_{Z_t}$ for all $t, t' \in \mathbb{N}$: the observations are independent draws from a fixed distribution P. The Bayes-optimal hypothesis is time-invariant, meaning $h_t^* = h^*$ for all t, and a time-agnostic hypothesis suffices.

Prospective learning recovers PAC learning as this special case. The prospective Bayes risk R_t^* is almost surely equal to the classical Bayes risk of P and does not depend on t. The hypothesis class $\mathcal{H}_{\text{ta}}$ of constant sequences is consistent, so Theorem 3.4.1 applies with the time-agnostic ERM learner. In other words, prospective learning asks nothing new of the learner when time carries no information.

One subtlety is worth noting. Even in the IID case, prospection can sometimes help. If a Bayesian learner has a misspecified prior, a prospective variant, one that models the rate of change of the posterior estimate and extrapolates forward, can converge to Bayes risk faster than the MAP estimator. This observation, established for the Bernoulli case, is minor in its practical implications but conceptually illuminating: even when the world is static, reasoning about how one's beliefs evolve over time can be beneficial (see De Silva, Ramesh, and Yang et al. (2024, Appendix B.1) for the

detailed example).

3.5.2 Scenario 2: Independent but not identically distributed data

The observations Z_t are independent across time, but the marginal distribution P_{Z_t} depends on t. The alternating-tasks example of Sections 2.4 and 3.3 is the canonical instance: $P_{Z_t} = P_A$ for odd t, $P_{Z_t} = P_B$ for even t. More generally, the distribution could cycle through a finite set of tasks with a known or unknown period, or change according to any pattern that is independent across time but varies in its marginals.

This scenario is where the force of prospective learning first becomes apparent. A time-agnostic learner cannot distinguish between odd and even times and, as Proposition 1 showed, can fail to outperform chance.A prospective learner, one whose hypothesis varies with t, can learn separate classifiers for each phase and achieve Bayes risk.

The periodic sub-case admits exact sample complexity analysis, as shown in Section 3.4: for a period-T process, learning reduces to T independent ERM problems, and the total sample complexity grows linearly in T. When the period is unknown, the learner can search over candidate periods using the countable union $\bigcup_{T\in\mathbb{N}} \mathcal{H}_T$, and Theorem 3.4.2 guarantees strong learnability.

Scenario 2 connects most directly to multi-task learning, where the learner has access to data from several tasks and seeks to perform well on each, and to online meta-learning, where tasks arrive sequentially and the learner adapts using a shared inductive bias. The key difference is that in prospective learning, the learner is not told which task is active at any given time and must instead infer the task identity from the temporal

pattern. This makes the problem harder but also makes it possible to *anticipate* future tasks rather than merely *adapt* to them after they arrive.

3.5.3 Scenario 3: Data that is neither independent nor identically distributed

The most general supervised setting. The stochastic process Z has arbitrary dependence across time: Z_t can depend on $Z_{t-1}, Z_{t-2}, \ldots$ in any way. Concrete instances include Markov chains, hidden Markov models (HMMs), and non-stationary processes with hierarchical structure.

The introduction of dependence creates a new consideration. If the process has an invariant distribution, as any ergodic Markov chain does, then the time-averaged prospective loss $\bar{\ell}_t(h, Z)$ converges to the expected loss under the invariant distribution for any hypothesis, making the time-averaged Bayes risk trivially achievable by a time-agnostic learner. In such cases prospective learning in the time-averaged sense is vacuous: there is nothing to prospect.

The resolution is to use a discounted loss, which weights near-future outcomes more heavily. With a discount factor $\gamma \in [0, 1)$, the prospective loss becomes

$$\bar{\ell}_t(h, z; \gamma) = (1 - \gamma) \sum_{s=t+1}^{\infty} \gamma^{s-t-1} \ell(s, h_s(x_s), y_s),$$

and the Bayes risk is no longer trivial. For a two-state Markov chain with self-transition probability θ and discount factor γ, the Bayes risk can be computed analytically and

depends on both parameters. A prospective learner that estimates the transition matrix from data and uses it to predict the state distribution at each future time converges to this Bayes risk; a time-agnostic learner, which cannot model the dependence, performs at chance.

Scenario 3 connects to task-agnostic continual learning, where the learner faces a stream of data without being told when or whether the task changes. But the goals differ sharply. Continual learning typically seeks to avoid catastrophic forgetting, that is, retaining performance on past tasks while learning new ones. Prospective learning seeks to *predict future performance*, learning the dynamics of how tasks change so as to anticipate what comes next. A continual learner adapts; a prospective learner anticipates. The foraging experiments of Section 3.7 will give a concrete example of this distinction.

3.5.4 Scenario 4: The future depends on the learner's predictions

In the three scenarios above, the stochastic process Z evolves independently of the learner's hypothesis. The learner observes and predicts but does not influence. Scenario 4 relaxes this: the distribution of Z_{t+1} can depend on the learner's prediction $h_{t+1}(X_{t+1})$ at time $t+1$.

The canonical instance is a Markov decision process (MDP) in which the learner's prediction is an action that influences the next state. Consider a two-state process with $P(Y_{t+1} = j \mid Y_t = j', h_{t+1}(1) = k) = \theta_k$ if $j = j'$ and $1 - \theta_k$ otherwise. The learner's "prediction" h_{t+1} is simultaneously a decision: it determines the transition dynamics. The Bayes-optimal hypothesis sequence must balance exploiting the current

state (choosing the action that yields high immediate reward) with exploring future states (choosing the action that yields high long-run return).

This scenario connects directly to reinforcement learning, but with important distinctions that will be developed in Chapter 4. Classical RL assumes stationary dynamics, episodic resets, and the Markov property; prospective learning with control, as formalized in the next chapter, relaxes all three. The MDP assumption is not required, as the process can be non-Markov. The environment can change over time. And the agent lives a single continuous life without resets.

This section does not develop Scenario 4 in full; its formalism requires extending the hypothesis to map states to actions rather than inputs to outputs, and the loss must account for the feedback loop between predictions and future distributions. These extensions are the subject of Chapter 4. The purpose of mentioning Scenario 4 here is to establish that it is a natural special case of the prospective learning framework, the most general one, and that the progression from Scenario 1 through Scenario 4 is a progression in the richness of the temporal structure the learner must model.

3.5.5 Beyond finite task families: the infinite-task setting

The scenarios above, with the exception of the IID case, implicitly assume a finite set of underlying tasks that recur in some pattern. But there is no reason to restrict attention to finite families. A natural extension is a process in which the distribution drifts continuously, so that the learner faces a genuinely novel distribution at every time step.

Consider a one-dimensional binary classification problem in which the input distribution drifts linearly: $X_t \sim P_t$ with P_t shifted by ϵt relative to P_0, and the label depends on whether x_t exceeds a threshold that also drifts with t. No finite collection of classifiers can be Bayes-optimal at all times; the optimal classifier at time t depends on t through the drift.

A prospective MLP with a Fourier time embedding succeeds on periodic processes but fails on this one, as the Fourier basis cannot represent a linear drift. A monomial embedding $\varphi_m(t) = (t, t^2, \ldots, t^d)$ succeeds on the linear-drift process but fails on periodic ones. This observation, established empirically, is significant for two reasons.

First, it shows that prospective learning can handle infinite-task settings: the formalism and the learner both extend beyond the periodic case without modification.

Second, it reveals that the time embedding φ is not a free lunch. The embedding must match the temporal structure of the stochastic process: Fourier for periodic dynamics, polynomial for polynomial drift, and presumably other bases for other structures. When the temporal structure is unknown a priori, the choice of φ becomes a model-selection problem. This is an open question with both theoretical and practical significance: is there a universal embedding, or must it always be process-specific? The question is revisited in Chapter 5.

3.5.6 The taxonomy as a unifying lens

The five settings above, namely IID, independent non-identical, dependent, action-dependent, and infinite-task, are not an ad hoc classification. They are organized by

a single axis: the amount of temporal structure the learner must exploit.

At one extreme (Scenario 1), time carries no information and classical learning suffices. At the other extreme (Scenario 4), time determines the dynamics of the environment and the learner must anticipate these dynamics to act optimally. In between, the intermediate scenarios correspond to increasing amounts of temporal structure: shared marginals but varying conditionals (Scenario 2), temporal dependence (Scenario 3), continuous drift (infinite-task).

Each known learning paradigm sits at a specific point on this axis:

- **PAC learning** is Scenario 1: time is absent.
- **Domain adaptation and transfer learning** are a two-time-point special case of Scenario 2: the learner has data from a source distribution and must predict on a target distribution, with at most one shift.
- **Multi-task and meta-learning** are Scenario 2 with a finite task family: the learner sees tasks in sequence and adapts, but cannot predict which task comes next (because tasks are drawn IID from a meta-distribution).
- **Task-agnostic continual learning** is Scenario 3 without dynamics modeling: the learner sees a non-stationary stream and adapts retrospectively, without anticipating future changes.
- **Reinforcement learning** is Scenario 4 under the MDP assumption: the learner acts in a stationary, Markov, episodic environment.

Prospective learning generalizes each of these by removing the specific structural assumption that restricts the paradigm to its position on the axis. The unifying claim, that all of these paradigms are special cases of a single time-indexed framework, is the subject of Chapter 5.

3.6 Empirical validation

The preceding sections developed the theory: prospective ERM, implemented by feeding a time embedding to a standard architecture, is a strong prospective learner under consistency and concentration conditions. This section asks whether the theory works in practice. The answer, across synthetic, MNIST, and CIFAR-10 experiments spanning Scenarios 2 and 3, is that it does: prospective ERM converges to near-Bayes prospective risk while a range of online, continual, and classical learning baselines plateau at or near chance.

The section is organized around four questions. First, does prospective ERM achieve low prospective risk on canonical problems? Second, how do existing methods, designed for non-stationary data but not for prospection, compare? Third, how robust is prospective ERM to relaxations of the idealized setup? Fourth, does the choice of time embedding matter?

3.6.1 Experimental setup

All experiments follow a common protocol. A stochastic process generates a sequence of 50,000 samples. For each time t in a grid of evaluation points, a learner is trained on data $z_{\leq t}$ and its prospective risk $R_t(\hat{h})$ is estimated by averaging the per-step loss

over the remaining samples $z_{>t}$, which serves as a Monte Carlo approximation to the conditional expectation. Results are averaged over five random seeds governing both the sample sequence and the network initialization. Error bars show one standard deviation.

Architecture and training. The prospective ERM learner uses a multi-layer perceptron (MLP) for synthetic and MNIST tasks, and a convolutional neural network (CNN) for CIFAR-10. In both cases, the network receives a d-dimensional Fourier time embedding $\varphi_f(s) = (\sin(\omega_1 s), \ldots, \cos(\omega_{d/2} s))$ with $\omega_i = \pi/i$ and $d = 50$, concatenated to the input (for MLPs) or added to the output of the convolutional layers (for CNNs). All networks are trained with SGD, Nesterov momentum, cosine-annealed learning rate, and cross-entropy loss. Models for different evaluation times t are trained independently, with no warm-starting or parameter sharing across t.

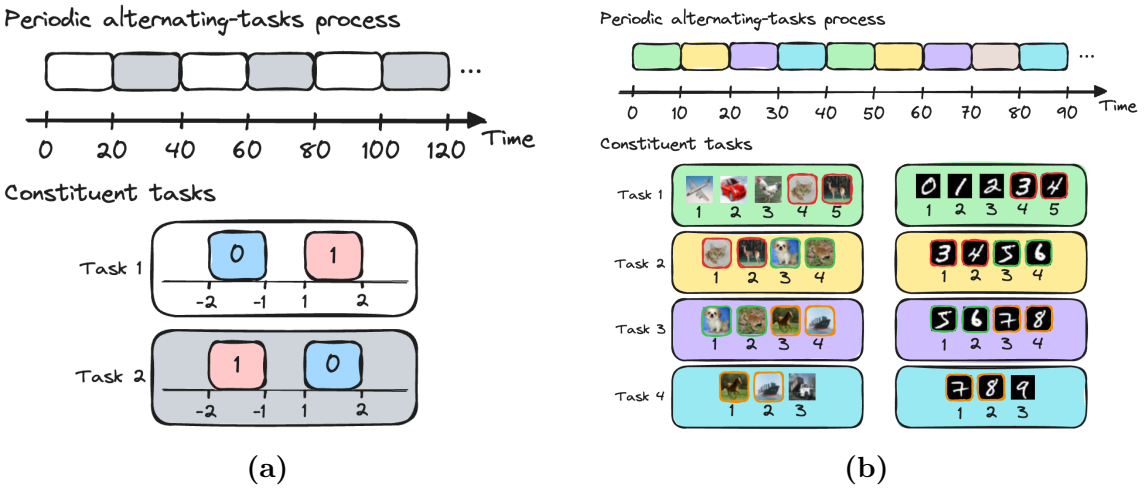
Baselines. Four baselines are considered, each representing a different existing approach to learning under non-stationary data.

Follow-the-Leader minimizes the empirical risk over all past data. It is the classical no-regret online learning algorithm, implemented here in batch mode: at each evaluation time t, a fresh network is trained on $z_{\leq t}$. Its output is a time-agnostic hypothesis.

Online SGD fine-tunes the network in a streaming fashion. At each time step, the network's weights are updated once using the most recent eight samples. The hypothesis at any moment is the current state of the network, making it time-agnostic by construction.

Bayesian gradient descent is a task-agnostic continual learning algorithm designed for settings where the learner does not know when or whether the task changes. It maintains an approximate posterior over the network weights and updates it online. Like online SGD, it produces a time-agnostic hypothesis at any given moment.

Chance is the constant predictor $\hat{y} = \mathbb{E}[Y]$. Its prospective risk is 1/2 for binary classification on the synthetic tasks and 0.742 for the multi-class MNIST and CIFAR-10 tasks.



(a) **(b)**

Figure 3.3: (a) Synthetic process for Scenario 2. The process alternates every 20 time steps between Task 1 (green), where the label is the sign of the input, and Task 2 (yellow), where the label is the negative of the sign. The Bayes-optimal classifier for one task is maximally wrong on the other. (b) CIFAR-10 (left) and MNIST (right) processes for Scenario 2. The process cycles every 10 time steps through four tasks (green, yellow, purple, turquoise), each defined on an overlapping subset of classes with independently assigned labels — shown beneath each image (CIFAR-10) or original digit (MNIST). Classes appearing in multiple tasks receive different labels across tasks, ensuring no fixed classifier achieves low risk throughout the cycle.

Stochastic processes for Scenario 2. For each dataset, we construct a process in which data is independent across time but the marginal distribution cycles through a

fixed set of tasks. For the synthetic data, two one-dimensional binary classification tasks are defined on the input domain $[-2,-1] \cup [1,2]$: in Task 1 the label is the sign of the input, and in Task 2 the label is the negative of the sign. The process alternates between these two tasks every 20 time steps (Fig. 3.3a). For MNIST and CIFAR-10, four tasks are constructed from the original ten classes by partitioning them into overlapping groups: Task 1 uses classes 1–5 with labels 1–5, Task 2 uses classes 4–7 with labels 1–4, Task 3 uses classes 6–9 with labels 1–4, and Task 4 uses classes 8–10 with labels 1–3. Classes that appear in multiple tasks receive different labels in each, so for instance, an image of class 6 is labeled 3 in Task 2 and 1 in Task 3. The process cycles through the four tasks in order, switching every 10 time steps (see Fig. 3.3b). In all three settings, the Bayes-optimal classifier changes at each task switch, ensuring that no time-agnostic hypothesis can achieve low prospective risk.

Stochastic processes for Scenario 3. For each dataset, we construct a process in which the task sequence is governed by a hierarchical hidden Markov model (see Fig. 3.4), introducing temporal dependence between successive observations. The four tasks are the same as in the Scenario 2 experiments. The hierarchy operates as follows: the outer process partitions the four tasks into two pairs, Tasks 1 and 2 in one group and Tasks 3 and 4 in the other, and switches between groups every 10 time steps. Within each group, an inner Markov chain with self-transition probability 0.8 governs which of the two tasks is active at each step. For the synthetic data, the same hierarchical structure is used with four two-dimensional binary classification tasks whose class boundaries differ across quadrants of the input space. The combination of outer switching and inner Markov dynamics ensures that the resulting process has no

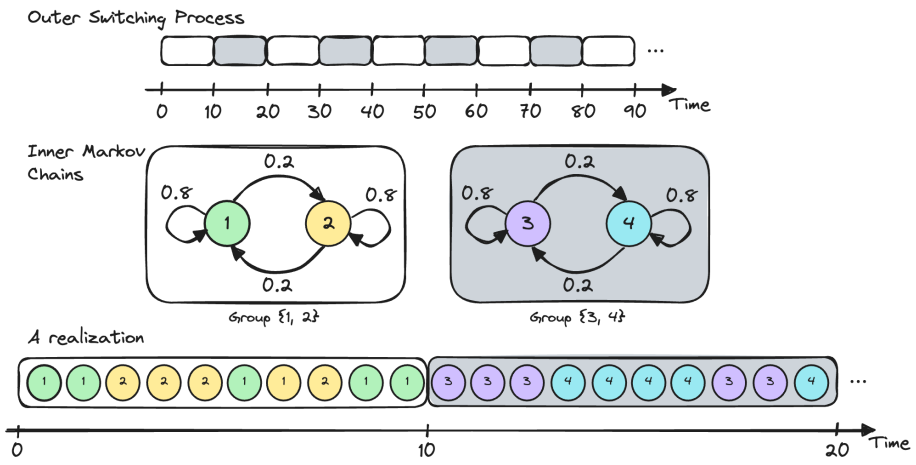


Figure 3.4: Scenario 3: a hierarchical hidden Markov model governs the task sequence. An outer deterministic process switches every 10 time steps between two groups of tasks. Within each group, an inner Markov chain with self-transition probability 0.8 governs which task is active at each step. The resulting process has no stationary distribution, since the long-run frequency of each task depends on where in the outer cycle the process is observed. Unlike Scenario 2, the task sequence within each block is stochastic and cannot be predicted from time alone; the learner must model the Markov dynamics.

stationary distribution: the long-run frequency of each task depends on when in the outer cycle the process is observed. This non-stationarity is what makes prospective learning non-trivial, as a time-agnostic learner that averages over the past cannot converge to the correct classifier at any given moment.

3.6.2 Prospective ERM achieves low prospective risk

Scenario 2 (independent, non-identical). The task alternates periodically: for synthetic data, two one-dimensional binary classification tasks switch every 20 time steps; for MNIST and CIFAR-10, four overlapping-class tasks cycle every 10 time steps.

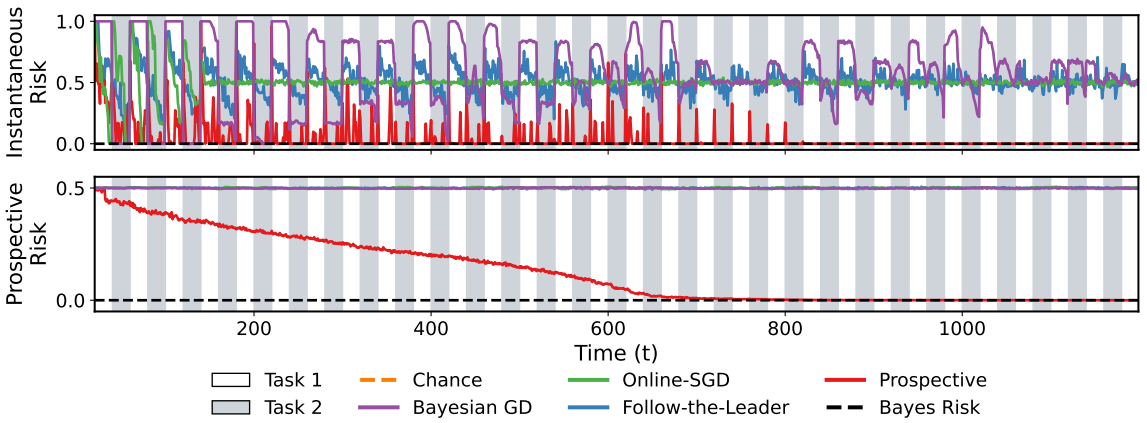


Figure 3.5: Instantaneous and prospective risk over time for Scenario 2 (synthetic data), averaged over 5 random seeds. The alternating white and gray background bands mark the periodic task switches (every 20 time steps). The online and continual learning baselines exhibit sharp risk spikes at task switches, whereas the prospective learner quickly dampens these spikes and drives both instantaneous and prospective risk toward zero.

On all three datasets, the prospective risk of prospective ERM drops rapidly toward Bayes risk as t increases (see Figs. 3.5 and 3.6). On synthetic data, where Bayes risk is zero, the prospective risk reaches zero by around $t = 400$. On MNIST, it drops below 0.1 by $t = 2000$. On CIFAR-10, which is the hardest setting, where a CNN must learn four different multi-class classification problems and their temporal pattern, the risk drops below 0.2 by $t = 15{,}000$.

Scenario 3 (dependent, non-stationary). The task sequence is governed by a hierarchical hidden Markov model: an outer process switches every 10 steps between two inner Markov chains, each governing transitions between two of the four tasks. The hierarchy ensures that the process has no stationary distribution, so a time-agnostic hypothesis cannot converge to Bayes risk by averaging.

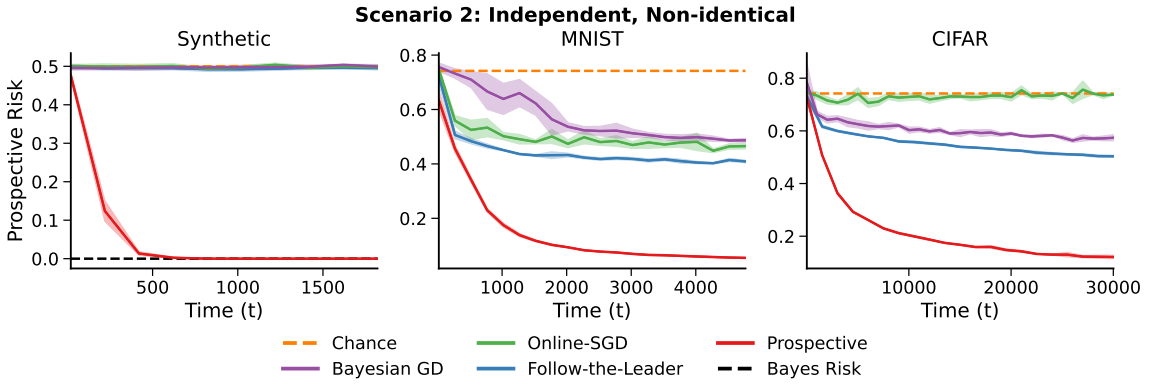


Figure 3.6: Prospective risk for Scenario 2 across synthetic, MNIST, and CIFAR-10 tasks. Prospective learners approach Bayes risk in all three settings, while all other baselines fail to achieve low prospective risk.

Again, prospective ERM converges to near-Bayes risk on all three datasets (see Fig. 3.7). The convergence is slower than in Scenario 2, as expected, since the learner must estimate Markov transition probabilities in addition to the task-specific classifiers. But the qualitative story is the same: given enough data, the prospective learner achieves risk close to the optimum.

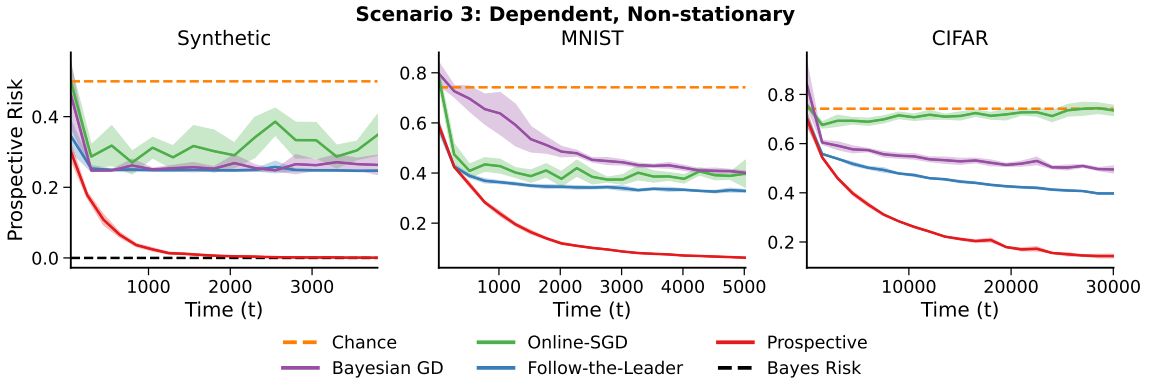


Figure 3.7: Prospective risk for Scenario 3 across synthetic, MNIST, and CIFAR-10 tasks. Prospective learners approach Bayes risk in all three settings, while all other baselines fail to achieve low prospective risk.

3.6.3 Baselines fail to prospect

Across all experiments, the three baseline methods, Follow-the-Leader, online SGD, and Bayesian gradient descent, achieve prospective risk far worse than prospective ERM.

Follow-the-Leader performs at or near chance in every Scenario 2 and Scenario 3 experiment. This is predicted by Proposition 3.3.1: Follow-the-Leader outputs a time-agnostic hypothesis, and the flipped-labels construction shows that no time-agnostic hypothesis can beat chance on alternating tasks with complementary class labels. The experimental tasks are not as extreme as the flipped-labels construction, since the classes overlap rather than being perfectly complementary, but the structural limitation is the same.

Online SGD shows transient improvements immediately after each task switch, as it fine-tunes toward the new task, but these improvements are lost at the next switch. Its prospective risk oscillates around chance. This is the expected behavior: online SGD tracks the current distribution but does not model the dynamics of how distributions change, and therefore cannot anticipate future switches.

Bayesian gradient descent performs comparably to online SGD. Its posterior updates provide some memory of past tasks, but without a model of temporal dynamics, this memory does not translate into anticipation.

Among the three baselines, Follow-the-Leader is notably the best performer on some tasks. This is initially surprising given that it is the simplest method, but reflects the

fact that it is trained in batch on all past data, while online SGD and Bayesian gradient descent are updated in a streaming fashion and are more sensitive to hyperparameter choices and optimization dynamics. The comparison underscores that the gap between baselines and prospective ERM is not an artifact of poor baseline implementation; it is a structural gap between retrospective and prospective approaches.

3.6.4 Robustness to irregular sample arrivals

The theoretical analysis assumes that the learner receives exactly one sample per time step. In practice, data may arrive irregularly, sometimes many observations at once and sometimes none. Does prospective ERM still work?

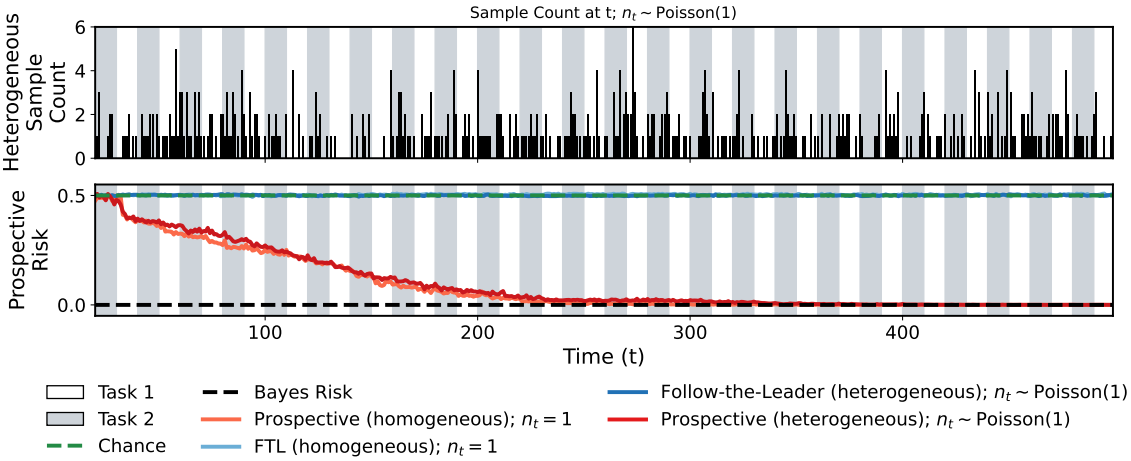


Figure 3.8: Prospective risks over time of Follow-the-Leader (FTL, blue) and prospective learner (red) trained on homogeneously (lighter shade) and heterogeneously (darker shade) sampled data from the periodic process in Fig. 3.3a. Homogeneous sampling is where you get exactly one sample each time step. In heterogeneous sampling, there can be missing samples and/or multiple samples available per time step.

When the number of samples per time step is drawn from a Poisson distribution with mean $\lambda = 1$, so that some time steps have zero observations and others have

several, prospective ERM achieves prospective risk comparable to the homogeneous (exactly-one-per-step) setting as depicted in Fig. 3.8. Follow-the-Leader, by contrast, remains at chance in both cases. This result suggests that the framework is not fragile to the sampling schedule. The key requirement is that the learner sees data from enough distinct time steps to estimate the temporal pattern, not that it sees exactly one sample per step.

3.6.5 The time embedding matters and can fail

Perhaps the most practically significant empirical finding is that the choice of time embedding φ is not interchangeable. The Fourier embedding φ_f, used in all the experiments above, is effective for periodic processes. It is not effective for processes with non-periodic temporal structure.

Consider the infinite-task linear-drift process described in Section 3.5: the input distribution shifts by ϵt at each time step, and the decision boundary drifts accordingly as illustrated in Fig. 3.9a. No finite set of classifiers is optimal; the learner faces a genuinely novel task at every time. A prospective MLP with the Fourier embedding φ_f fails on this problem, as shown in the bottom row of Fig. 3.9b, because the Fourier basis cannot represent a linear function of time and its prospective risk does not converge. A monomial embedding $\varphi_m(t) = (t, t^2, \ldots, t^d)$ succeeds: the network learns the linear relationship between time and the decision boundary and achieves near-Bayes risk.

The converse also holds. On the periodic alternating-tasks process, the monomial embedding performs poorly, since polynomials are a bad basis for periodic functions, while the Fourier embedding succeeds, as shown in Fig. 3.9b.

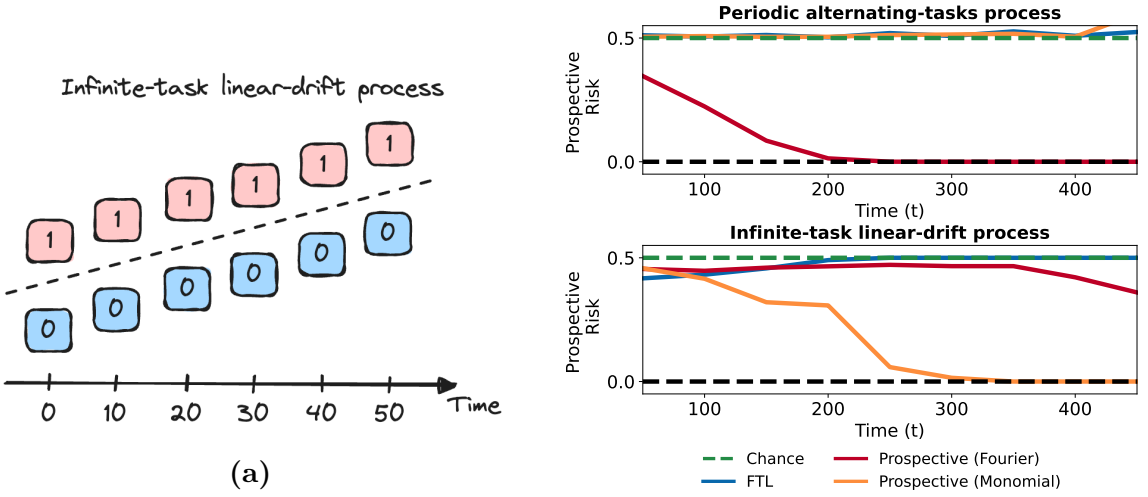


Figure 3.9: (a) Illustration of the infinite-task linear-drift process, where the decision boundary shifts continuously over time. (b) Prospective risk over time for the periodic alternating-tasks and linear-drift processes under Fourier and monomial time embeddings. Each embedding succeeds on the process that matches its inductive bias, Fourier for periodic tasks and monomial for linear drift, and fails on the other, demonstrating that the choice of time embedding is a critical modeling decision.

This complementarity has a clear implication. The time embedding is a modeling choice, analogous to the choice of kernel in kernel methods or the choice of architecture in deep learning. It encodes the learner's inductive bias about the temporal structure of the stochastic process. A mismatch between the embedding and the process can cause the learner to fail even when the data is abundant and the architecture is expressive. In Section 3.5 this was noted as an open question, namely whether a universal embedding exists, and the empirical evidence here suggests the answer may be no, at least within the class of fixed-basis embeddings.

3.6.6 Summary

The experiments of this section establish three things. First, prospective ERM works in practice: it converges to Bayes risk on problems spanning two scenarios, three datasets, and two architectures, while baselines from the online, continual, and classical learning literature do not. Second, the framework is robust to practical relaxations including heterogeneous sampling and online training. Third, the time embedding is a consequential design choice that can determine success or failure. Taken together, these findings validate the theory of Section 3.4 and motivate the extensions to sequential decision-making in Chapter 4.

3.7 Beyond prediction: a first look at prospective decision-making

The experiments of Section 3.6 validated prospective ERM on supervised learning problems spanning Scenarios 2 and 3. In all of these, the learner observes and predicts but does not act, as its hypothesis $h_t(x_t)$ produces a label $\hat{y}_t$ that has no influence on future data. This section crosses that boundary. It introduces a sequential decision-making problem, a prospective foraging task, and shows that the same principle that made prospective ERM successful in supervised learning, namely making time an input, transforms the performance of a reinforcement learning agent.

3.7.1 The foraging environment

The environment is a one-dimensional linear track with seven positions ($\mathcal{S} = \{0, 1, \ldots, 6\}$) and two reward patches, A at position 1 and B at position 5, separated by three cells.

At each time step, the agent can move one cell left, one cell right, or stay in place ($\mathcal{A} = \{-1, 0, +1\}$, clipped to the track boundaries). Rewards alternate between the two patches with period $2N = 20$ time steps: during the first $N = 10$ steps of each cycle, patch A is active and its reward decays exponentially from a peak value while patch B yields zero; during the next N steps, the roles reverse. The agent observes only the reward at its current position, and there are no resets, as the agent lives a single continuous life.

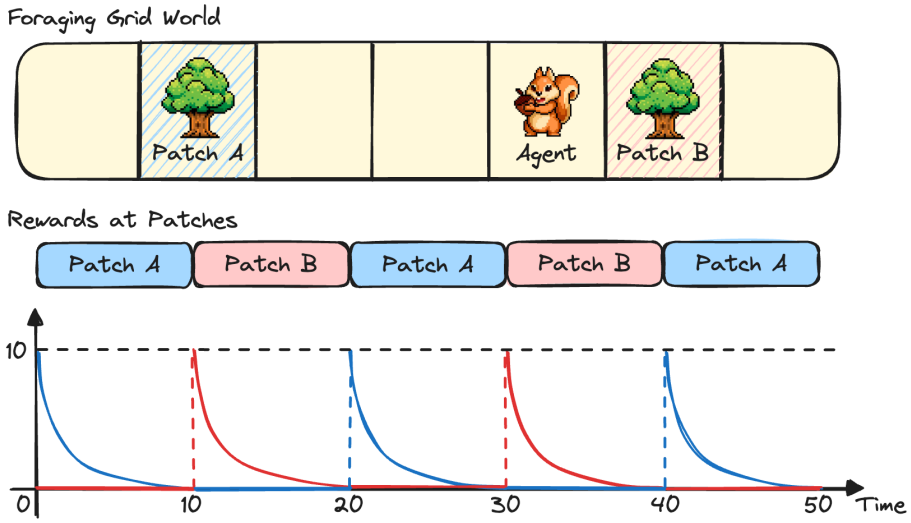


Figure 3.10: A schematic diagram of the foraging environment including the time-varying rewards at the two patches.

We can derive the optimal policy for this task (see Section C.1 for the derivation) and it requires genuine prospection. When the active patch's reward has decayed sufficiently, the agent must leave *before* the switch occurs, travel to the other patch, and arrive in time to harvest the peak reward at the moment of activation. This means accepting zero immediate reward during travel, a sacrifice that a purely retrospective agent optimizing the current reward would never make.

3.7.2 Time-aware fitted Q-iteration

A natural baseline for this problem is fitted Q-iteration (FQI) (Ernst et al., 2005; Munos and Szepesvari, 2008), a batch reinforcement learning algorithm that learns an action-value function $Q(s, a)$ from observed transitions and uses it to select actions greedily. In its standard form, FQI is time-agnostic: the Q-function depends on the state and action but not on time. This means it learns a single policy that is deployed identically at every moment, which is exactly the time-agnostic hypothesis that Proposition 3.3.1 showed to be structurally insufficient for non-stationary processes.

The prospective modification is minimal. Instead of learning $Q(s, a)$, the agent learns a time-indexed Q-function $Q(s, a, t)$ that takes a time embedding $\varphi(t)$ as an additional input alongside the state and action. The Q-function is implemented as a regression model, either a neural network or a random forest, that receives $(s, a, \varphi(t))$ and predicts the expected discounted return from taking action a in state s at time t. The policy derived from this Q-function is also time-aware: $\pi(s, t) = \arg\max_a Q(s, a, t)$, selecting different actions at different times even in the same state.

This modification is the decision-making analog of the supervised prospective ERM recipe: take an existing algorithm, augment its inputs with a time embedding, and train on time-stamped data. The Q-function $Q(s, a, t)$ is a time-indexed hypothesis in exactly the sense of Section 3.2, a function that takes time as an input alongside its usual arguments.

3.7.3 Results

Two agents are compared: standard (time-agnostic) FQI and time-aware FQI with the time embedding. To compare agents against the Bayes-optimal policy on a common scale, we report the normalized prospective regret: the difference between the oracle agent's cumulative reward and the learner's cumulative reward over a finite evaluation horizon, divided by the oracle's cumulative reward. A normalized regret of zero means the agent matches the oracle; a value of one means it collects no reward. This metric is analogous to the prospective risk of Section 3.2 but expressed in terms of reward rather than loss and normalized to account for the absolute scale of the oracle's return.

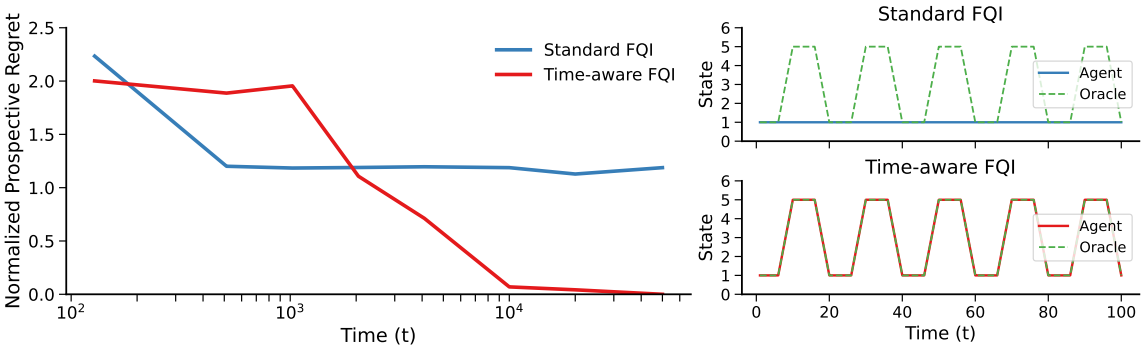


Figure 3.11: Prospective decision-making in the foraging environment. **Left:** Normalized prospective regret as a function of training time for standard (time-agnostic) FQI and time-aware FQI. The time-aware agent converges to near-zero regret, matching the oracle policy, while the time-agnostic agent plateaus at high regret. **Right, top row:** State sequence of the time-agnostic FQI agent over 100 time steps after training on 50,000 interactions. The agent remains at position 1 for the entire duration, never leaving the patch, as it cannot represent the fact that the value of staying depends on time. **Right, bottom row:** State sequence of the time-aware FQI agent over the same 100-step window. The agent's movement pattern aligns with the Bayes-optimal state sequence: it departs the active patch before the reward switch, travels to the alternative patch, and arrives at the moment of activation. The contrast illustrates that making time an input to the Q-function transforms the agent from one that exploits the present to one that anticipates the future.

The time-agnostic agent learns a fixed policy that stays at whichever patch currently yields reward (see Fig. 3.11). It never learns to leave a yielding patch, because the immediate cost of travel always outweighs the immediate benefit. Its prospective risk plateaus well above Bayes risk.

The time-aware agent, FQI with the time embedding, performs substantially better. Because $Q(s, a, t)$ can represent the fact that the reward at patch A is high at certain times and low at others, the derived policy can learn to depart before the switch. In practice, this agent converges to near-optimal behavior, reproducing the qualitative pattern of the Bayes-optimal policy: remain at the active patch while its reward is high, depart before the switch, travel to the alternative, and arrive at the moment of activation.

The contrast between the two agents is the same contrast that Proposition 3.3.1 identified in the supervised setting, now manifested in control: a time-agnostic Q-function cannot represent the fact that the value of an action depends on *when* it is taken, and consequently the derived policy cannot anticipate the reward reversal. A time-aware Q-function can.

3.7.4 This is a proof of concept, not a complete formalization

The time-augmented FQI agent described above is adapted heuristically. Several subtleties are glossed over. The agent observes rewards only at its current position, and rewards at unvisited positions must be imputed. The discount factor and planning horizon interact with the reward periodicity in ways that affect convergence. The training objective is the standard Bellman backup, not the prospective empirical risk

of Section 3.4. And the theoretical guarantees of Theorem 3.4.1 do not directly apply, because the learner's actions influence the future data distribution, a feedback loop that the supervised framework does not address.

Chapter 4 addresses these subtleties head-on. It provides a formal definition of prospective learning with control, replaces the Bellman-based training objective with an estimation-theoretic one, introduces a dedicated algorithm (PLC) that combines instantaneous and cumulative loss regressors with short-horizon planning, and proves a learnability theorem analogous to Theorem 3.4.1. The purpose of the present section is to establish a single claim: the principle of making time an input, which transformed supervised prospective learning from impossible (Proposition 3.3.1) to Bayes-optimal (Theorem 3.4.1), has the same effect in sequential decision-making.

3.8 Chapter summary and bridge to Chapter 4

This chapter developed the formalism of prospective learning and established its central result. Data is modeled as a stochastic process; the learner's output is a time-indexed hypothesis sequence; and the learner's performance is measured by the prospective risk, a conditional expectation of the time-averaged future loss given the realized past. Time-agnostic ERM, the learner used by all classical approaches, provably fails on stochastic processes where the optimal hypothesis varies with time. Prospective ERM, which differs from time-agnostic ERM only in that time is provided as an additional input, is a strong prospective learner under consistency and uniform concentration conditions. The result holds for neural networks and decision forests alike, is robust to heterogeneous sampling and online training, and hinges on a design choice, the

time embedding, that must match the temporal structure of the problem.

The one setting this chapter deferred is Scenario 4: problems where the learner's predictions influence the future. The foraging experiment of Section 3.7 showed that an agent trained via time-aware fitted Q-iteration, equipped with a time embedding, can solve a sequential decision-making problem that defeats retrospective agents. But the experiment was a proof of concept, not a formal treatment. Chapter 4 provides the formal treatment. It extends the prospective learning framework to a setting where the hypothesis maps states and times to actions, the loss accounts for the feedback loop between actions and future states, and the learner must impute rewards for actions it did not take. The result is a framework called prospective learning with control, the most general instantiation of the prospective learning program, together with a learnability theorem and an algorithm that achieves near-optimal performance orders of magnitude faster than reinforcement learning baselines.

Chapter 4
Prospective Control

Chapter 3 developed prospective learning for settings where the learner observes and predicts but does not act. The hypothesis $h_t(x_t)$ produces a prediction $\hat{y}_t$, but $\hat{y}_t$ has no influence on the data the learner will encounter in the future. The foraging experiment of Section 3.7 showed that this boundary can be crossed: a time-aware variants of PPO (Schulman et al., 2017), SAC (Haarnoja, Zhou, Abbeel, et al., 2018), and FQI (Munos and Szepesvari, 2008), equipped with a time embedding, solved a sequential decision-making problem that outperformed their time-agnostic counterparts. But the treatment was informal, as the learner was adapted heuristically and several subtleties were deferred.

This chapter provides the formal treatment. It extends the prospective learning framework to a setting called *prospective learning with control (PLC)*, in which the learner's decisions at each time step influence the future state of the environment. The extension requires three modifications to the Chapter 3 setup: the hypothesis must map states and times to actions rather than inputs to labels; the loss must account for the fact that the learner observes rewards only at visited states, not at all states; and the theory must handle the feedback loop between the learner's decisions and the future distribution of states.

The result is a framework grounded in estimation theory rather than in the Bellman lineage of classical reinforcement learning. The learner does not solve a dynamic

programming problem; it trains two regressors, one for instantaneous losses and one for cumulative future losses, and plans by enumerating short action sequences scored by these regressors. Under consistency and concentration conditions analogous to those of Theorem 3.4.1, the learner achieves Bayes-optimal prospective risk.

The chapter is organized as follows. Section 4.1 motivates the extension from prediction to action and contrasts it with classical RL. Section 4.2 defines the formalism. Section 4.3 states the learnability result. Section 4.4 describes the PLC algorithm. Section 4.6 presents experiments on a prospective foraging task. Section 4.7 positions the framework relative to existing work in reinforcement learning.

4.1 From prediction to action

In the prospective learning framework of Chapter 3, the stochastic process $Z = (Z_t)_{t\in\mathbb{N}}$ evolves independently of the learner. The learner's hypothesis $h_t(x_t)$ is a prediction, a guess at what y_t will be, and has no causal effect on Z_{t+1}. This is adequate for many problems: classification under distribution shift, forecasting under non-stationarity, pattern recognition in streaming data. But it excludes a large class of settings in which the learner is an agent whose decisions shape its own future.

Consider an animal foraging in an environment with two food patches whose yields alternate over time. The animal must decide, at each moment, whether to stay at the current patch or travel to the other. If it stays, it harvests whatever reward the current patch offers. If it travels, it incurs a period of zero reward but may arrive at the other patch just as its yield peaks. The optimal policy requires anticipating the

switch, leaving before the current patch is depleted and arriving at the alternative in time to exploit its peak. A purely retrospective agent, optimizing immediate reward, would never leave a yielding patch.

This is Scenario 4 from Section 3.5, made concrete. The key property is that the agent's action at time t, either stay or move, determines the state at time $t+1$, which in turn determines what rewards are available. The stochastic process is no longer exogenous; it is co-determined by the environment's dynamics and the learner's policy.

Classical reinforcement learning addresses this setting under three assumptions: the environment is a Markov decision process (MDP), the dynamics are stationary, and the agent experiences multiple episodes with resets to an initial state. Prospective learning with control relaxes all three. The process need not be Markov, as the reward at time t can depend on the full history and not just the current state. The dynamics can be non-stationary, meaning the reward function can change over time in structured ways. And the agent lives a single continuous life without resets, as real-world biological agents do.

The relaxation of the MDP assumption is conceptually significant. Classical RL inherits its theoretical apparatus (including value functions, Bellman equations, policy iteration) from dynamic programming (Bellman, 1954), which requires the Markov property. Without it, the Bellman recursion does not hold, and the standard RL toolkit does not apply. Prospective learning with control instead inherits its apparatus from estimation theory: the learner estimates losses as functions of state and time, and plans by evaluating candidate action sequences against these estimates. The

connection to Chapter 3 is direct, as the learner is still doing ERM with time as an input, but now the input includes the agent's own state and the label includes the consequences of the agent's actions.

4.2 Formalism

The setup extends Chapter 3's stochastic-process framework to include actions. The notation is chosen to be consistent with Chapter 3 while adopting the standard RL vocabulary of states, actions, and rewards.

State and action. At each time $t \in \mathbb{N}$, the agent occupies a state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$. The action determines the next state according to a (possibly non-stationary, possibly non-Markov) transition rule: s_{t+1} is a function of the history $(s_1, a_1, \ldots, s_t, a_t)$ and, potentially, of exogenous randomness. In the simplest case, which is the one treated in the experiments below, the dynamics are deterministic and depend only on the current state and action: $s_{t+1} = f(s_t, a_t)$.

Reward. At each time t, the environment generates a reward $r_t(s)$ for every state $s \in \mathcal{S}$. The agent observes only the reward at its current state: $r_t(s_t)$. Rewards at unvisited states are not observed. The reward function can be non-stationary, meaning r_t can depend on t in structured ways, and it evolves independently of the agent's actions. (However, the reward can also depend on both the state and action in general.) This is a key modeling choice: the environment changes over time (rewards shift), and the agent's actions determine *where* it is when those changes occur, but not *what* the changes are.

Hypothesis. A hypothesis (or policy) is a sequence $h = (h_1, h_2, \ldots)$ where each $h_t : \mathcal{S} \times \{t\} \to \mathcal{A}$ maps the current state and time to an action. Equivalently, h is a function $h : \mathcal{S} \times \mathbb{N} \to \mathcal{A}$. This is the direct analog of Chapter 3's time-indexed hypothesis $h : \mathcal{X} \times \mathbb{N} \to \mathcal{Y}$, with input space replaced by state space and output space replaced by action space.

Loss. The instantaneous loss at time t is defined as the negative reward at the agent's current state:

$$\ell(h_t(s_t, t),\, r_{t+1}) = -r_{t+1}(s_{t+1}),$$

where s_{t+1} is the state the agent reaches by executing action $h_t(s_t, t)$ from state s_t. The prospective loss at time t is a weighted cumulative future loss[1]:

$$\bar{\ell}_t(h, Z) = \sum_{k=t+1}^{\infty} w_{k-t} \cdot \ell(h_k(s_k, k),\, r_{k+1}),$$

where w_i is a non-negative weight sequence satisfying $\sum_{i=1}^{\infty} w_i < \infty$. In practice, a discounted weighting $w_i = \gamma^{i-1}(1-\gamma)$ is used. The prospective risk is the conditional expectation of the prospective loss given the realized history, as in Chapter 3:

$$R_t(h) = \mathbb{E}\left[\bar{\ell}_t(h, Z) \mid z_{\leq t}\right].$$

[1]Here, we use k as the summation index. In Chapter 3, we previously used s; this change avoids a notational conflict.

Counterfactual imputation. A difficulty that does not arise in Chapter 3's supervised setting is that the agent observes rewards only at visited states. To evaluate a candidate policy h on past data, the learner must estimate what reward the agent *would have received* had it visited a different state. Let $\tilde{r}_t(s)$ denote an estimate of $r_t(s)$ for states $s \neq s_t$ that were not visited at time t. The quality of this estimate affects the learner's ability to evaluate candidate policies, and consequently the sample efficiency of the algorithm. The PLC algorithm described in Section 4.4 addresses this by training a regressor that predicts instantaneous rewards as a function of state and time.

Prospective Bayes risk. As in Chapter 3, the benchmark is

$$R_t^* = \inf_{h \in \mathcal{F}_t} R_t(h),$$

the infimum over all $\mathcal{F}_t$-measurable policies. The Bayes-optimal policy is the one that, given the history $z_{\leq t}$, selects the sequence of future actions minimizing expected cumulative future loss. In the foraging example, this corresponds to the policy that departs the current patch at the optimal moment and arrives at the alternative patch precisely when its reward peaks.

4.3 Learnability

The main theoretical result for prospective control mirrors Theorem 3.4.1. The structure is identical: a consistency condition and a concentration condition together

imply that an ERM-style learner achieves Bayes-optimal risk.

Lemma 4.3.1 (existence of a good reference policy)**.** Consider a finite family $\mathcal{Z}$ of stochastic processes. Suppose there exists an increasing sequence of policy classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ with each $\mathcal{H}_t \subseteq (\mathcal{A}^{\mathcal{S}\times\mathbb{N}})^{\mathbb{N}}$ such that for all $Z \in \mathcal{Z}$,

$$\lim_{t\to\infty} \mathbb{E}\left[\inf_{h\in\mathcal{H}_t} R_t(h) - R_t^*\right] = 0,$$

where $h \in \mathcal{H}_t$ is measurable with respect to $\mathcal{F}_t$. Then there exists a sequence of $\mathcal{F}_t$-measurable policies $h^{(t)} \in \mathcal{H}_t$ whose empirical partial cumulative prospective loss converges to Bayes-optimal risk almost surely.

The proof follows the same Borel-Cantelli subsequence argument used in Theorem 3.4.1. The only additional complication is that the loss now involves imputed rewards $\tilde{r}_t(s)$ at unvisited states; the proof requires that the imputation error vanish asymptotically, which is ensured by the consistency of the instantaneous-loss regressor.

Theorem 4.3.1 (PLC is a strong prospective learner)**.** Under the conditions of Lemma 4.3.1, and the additional assumption that the gap between the true prospective loss and the maximum partial cumulative loss is bounded by a vanishing sequence ξ_t, the ERM-like minimizer

$$\hat{h}^{(t)} = \operatorname*{argmin}_{h\in\mathcal{H}_{i_t}} \max_{u_{i_t}\le m\le t} \sum_{s=1}^{m} w_s \cdot \tilde{\ell}(h_s(s_s, s), \tilde{r}_{s+1})$$

achieves Bayes-optimal risk asymptotically: $R_t(\hat{h}^{(t)}) - R_t^* \to 0$ in probability.

The theorem says, in essence, that the same recipe that worked for supervised prospective learning, ERM over time-indexed hypotheses with an expanding hypothesis class, works for control, provided the learner can estimate rewards at unvisited states well

enough. The finite-sample rate depends on the quality of the reward imputation and the complexity of the policy class; establishing explicit rates remains an open problem.

Proof. The proof combines three ingredients. First, Lemma 4.3.1 provides a reference policy whose empirical surrogate converges to Bayes risk. Second, the ERM minimizer $\hat{h}^{(t)}$ has empirical surrogate no worse than the reference policy by construction. Third, the concentration condition ensures that the empirical surrogate and the true prospective loss are uniformly close. The three ingredients compose in exactly the same way as in the proof of Theorem 3.4.1, with the imputed rewards $\tilde{r}$ playing the role that observed labels y_s played in the supervised setting. We refer the read to the full details given in Bai, Acharyya, and De Silva et al. (2025). □

4.4 The PLC algorithm

The theory of Section 4.3 establishes that an ERM-style learner over time-indexed policies achieves Bayes risk. This section describes a concrete algorithm, the Prospective Learner for Control (PLC), that implements this idea using two learned regressors and a short-horizon planning step inspired by Bertsekas (2023, Section 2.3.5) on Truncated Rollout with Terminal Cost Approximation.

4.4.1 Two regressors

PLC maintains two models, both receiving state-time pairs (s, t) as input:

The instantaneous-loss regressor $g_i(s, t; \theta)$ predicts the immediate loss (negative reward) the agent would incur at state s at time t. It is trained on observed state-time-

reward triples $(s_t, t, r_{t+1}(s_t))$ by minimizing mean squared error. Because the agent observes rewards only at visited states, g_i also serves as the counterfactual imputation mechanism: at planning time, $g_i(s, t)$ provides an estimate of the reward the agent would receive at any state s, whether or not the agent has visited s at time t.

The cumulative-loss regressor $g_c(s, t; \varphi)$ predicts the discounted cumulative future loss from state s at time t onward. It is trained on tuples $(s_t, t, \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k)$, the realized discounted return from the agent's actual trajectory, again by minimizing mean squared error.

The two regressors play complementary roles. The instantaneous regressor provides fine-grained, per-step loss estimates that are accurate at visited states and extrapolate to unvisited ones. The cumulative regressor provides a coarser but longer-horizon signal that captures the total cost of being in a given state at a given time. The ablation experiments in Section 4.6 show that using both regressors together yields substantially better performance than using either alone.

Both regressors receive a time embedding $\varphi(t)$ as part of their input, exactly as in Chapter 3's prospective ERM. The time embedding allows the regressors to capture the non-stationary structure of the reward function, for instance the periodic alternation of reward patches in the foraging task.

4.4.2 Planning

At each time step t, the agent must select the next action a_t. PLC does this by short-horizon planning: it enumerates all feasible action sequences of length H (the

planning horizon), scores each sequence using the two regressors, and selects the best.

For a candidate action sequence $(a_t, a_{t+1}, \ldots, a_{t+H-1})$, let $(s_t, s_{t+1}, \ldots, s_{t+H})$ be the resulting state trajectory under the known dynamics. The score of this sequence is

$$Q(s_{t:t+H}) = \underbrace{\sum_{h=1}^{H} \gamma^{h-1} g_i(s_{t+h}, t+h)}_{Q_{\text{finite}}} + \underbrace{\gamma^H \cdot g_c(s_{t+H}, t+H)}_{Q_{\text{terminal}}}.$$

The first term scores the sequence over the planning horizon using the instantaneous regressor. The second term estimates the cost-to-go beyond the horizon using the cumulative regressor. The agent selects the sequence with the lowest total score and executes its first action:

$$a_t^* = \text{first action of } \operatorname*{argmin}_{a_{t:t+H-1}} Q(s_{t:t+H}).$$

For small action spaces and short horizons, the foraging task has three actions and uses $H = 6$, giving $3^6 = 729$ candidate sequences, making exhaustive enumeration tractable. For larger problems, Monte Carlo tree search or other approximate planning methods can be substituted.

Need for a world model. The planning step as described assumes that the agent has access to the environment's transition dynamics, meaning that given a state s_t and an action sequence $(a_t, a_{t+1}, \ldots, a_{t+H-1})$, the agent can compute the resulting state trajectory $(s_{t+1}, \ldots, s_{t+H})$ without executing the actions. In the foraging environment

of Section 4.6 this assumption is satisfied because the dynamics are deterministic and known: moving left decrements the position, moving right increments it, and staying leaves it unchanged. In more complex settings, where dynamics are stochastic, high-dimensional, or unknown to the agent, this assumption must be relaxed. The natural extension is to replace the known dynamics with a learned forward model (or world model) $\hat{f}(s_t, a_t, t) \approx s_{t+1}$, trained alongside the two loss regressors on observed state transitions. The planning step would then enumerate candidate action sequences and roll them out under $\hat{f}$ rather than under the true dynamics. This introduces a compounding model-error problem, where errors in $\hat{f}$ accumulate over the H-step rollout, which is well-studied in model-based reinforcement learning and would apply equally here. The current work does not address this extension; developing PLC with a learned world model, and understanding how model error interacts with the prospective loss estimates, is an important direction for future work.

4.4.3 The full algorithm

The complete PLC procedure operates in two phases.

Warm-up phase. For the first T_{warmup} time steps, the agent follows a near-random policy, selecting actions uniformly at random. This phase serves two purposes: it populates the replay buffer $\mathcal{D}$ with state-time-reward triples from diverse states, and it provides initial training data for both regressors. The warm-up is necessary because the regressors cannot provide useful predictions, and therefore the planner cannot make good decisions, without some initial experience.

Online learning and planning phase. For each subsequent time step t:

1. Compute the cumulative discounted losses from the trajectory data in $\mathcal{D}$.
2. Update g_i to minimize MSE on observed instantaneous losses.
3. Update g_c to minimize MSE on computed cumulative losses.
4. Enumerate all feasible action sequences of length H from the current state s_t.
5. Score each sequence: $Q = Q_{\text{finite}} + Q_{\text{terminal}}$.
6. Select the sequence with the lowest score. Execute the first action.
7. Observe the instantaneous reward $r_{t+1}(s_{t+1})$ and store $(s_{t+1}, t+1, r_{t+1}(s_{t+1}))$ in $\mathcal{D}$.

The algorithm is online: both regressors are updated at every step using all data collected so far. The planning step is also online: it uses the current regressors to evaluate candidate sequences and selects the best one for the current moment. There is no separate training phase followed by a deployment phase; learning and acting are interleaved throughout the agent's life. The pseudocode of the complete PLC algorithm is given in Section C.5.

4.4.4 Relationship to Chapter 3

The PLC algorithm is recognizably a descendant of prospective ERM. The instantaneous regressor g_i is a prospective learner in the Chapter 3 sense: it takes (s, t) as input and predicts a loss, with time as an explicit input via the embedding $\varphi(t)$. The cumulative regressor g_c extends this to a longer horizon. The planning step, which

scores candidate action sequences against learned loss models, replaces the simple procedure of feeding $(t', x_{t'})$ to the network at inference time from Chapter 3 with a search over actions, but the core principle is the same: use time-aware models of the world to anticipate the future rather than react to it.

4.5 Experiments: prospective foraging

The PLC algorithm is evaluated on a prospective foraging task, the same environment introduced in Section 3.7, now treated with the full control formalism.

4.5.1 Environment

The environment is a one-dimensional linear track with seven positions ($\mathcal{S} = \{0, 1, \ldots, 6\}$) and two reward patches, A at position 1 and B at position 5, separated by three cells (see Fig. 4.1). At each time step, the agent can move one cell left, one cell right, or stay in place ($\mathcal{A} = \{-1, 0, +1\}$, clipped to the track boundaries).

Rewards alternate between the two patches with period $2N = 20$ time steps. During the first $N = 10$ steps of each cycle, patch A is active: its reward decays exponentially from a peak value, while patch B yields zero. During the next N steps, the roles reverse. The agent observes only the reward at its current position.

The environment has no resets i.e. the agent lives a single continuous life. This is a key property: there is no opportunity to "try again" with a fresh initial state. Every decision is permanent, and the consequences of a poor decision persist until the agent corrects course.

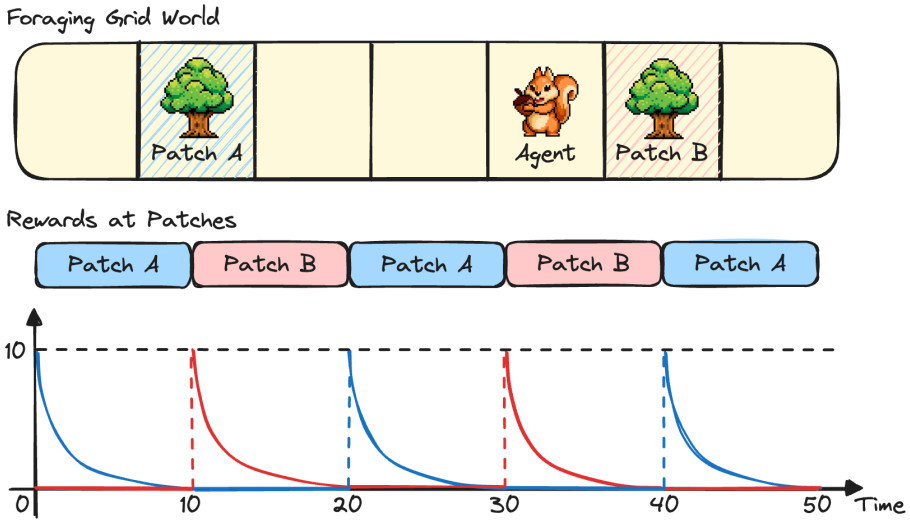


Figure 4.1: A schematic diagram of the foraging environment and time-varying rewards of the two patches.

The Bayes-optimal policy is known analytically and is derived in Section C.1. The agent should remain at the active patch while its reward is high, depart before the reward has fully decayed, travel to the other patch, and arrive at the moment of activation. This requires accepting zero reward during travel, a prospective sacrifice that no retrospective policy would make.

4.5.2 Baselines

PLC is compared against time-aware variants of three standard RL algorithms: Fitted Q-Iteration (FQI), Soft Actor-Critic (SAC), and Proximal Policy Optimization (PPO). The "time-aware" variants receive the same time embedding $\varphi(t)$ as PLC, ensuring that any performance difference is due to the algorithm, not to the presence or absence of temporal information. Time-agnostic versions of the same algorithms are also evaluated. We provide a detailed discussion on standard and time-aware variants of

FQI, SAC, and PPO in Sections C.2 to C.4 respectively.

The evaluation metric is *normalized prospective regret*: the difference between the agent's cumulative reward and the oracle (Bayes-optimal) agent's cumulative reward, normalized by the oracle's cumulative reward, averaged over a finite horizon. A regret of zero means the agent matches the oracle; a regret of one means it collects no reward.

4.6 Results

PLC converges orders of magnitude faster than RL baselines. PLC achieves near-zero normalized regret within approximately 100 time steps as demonstrated in Fig. 4.2. The time-aware RL baselines, FQI, SAC, and PPO with the same time embedding, eventually converge but require 10^3 to 10^5 time steps, one to three orders of magnitude more experience. Time-agnostic baselines plateau at suboptimal regret and never converge to the oracle.

The speed advantage of PLC is attributable to two factors. First, PLC's two-regressor architecture decomposes the problem into an instantaneous prediction task, which is relatively easy since the reward function is smooth in state and time, and a cumulative prediction task, which is harder but benefits from the instantaneous regressor's counterfactual estimates. Second, PLC's planning step explicitly searches over short action sequences, rather than learning a policy end-to-end as RL methods do. This decomposition into prediction and planning is more sample-efficient than the monolithic policy-optimization approach of SAC and PPO.

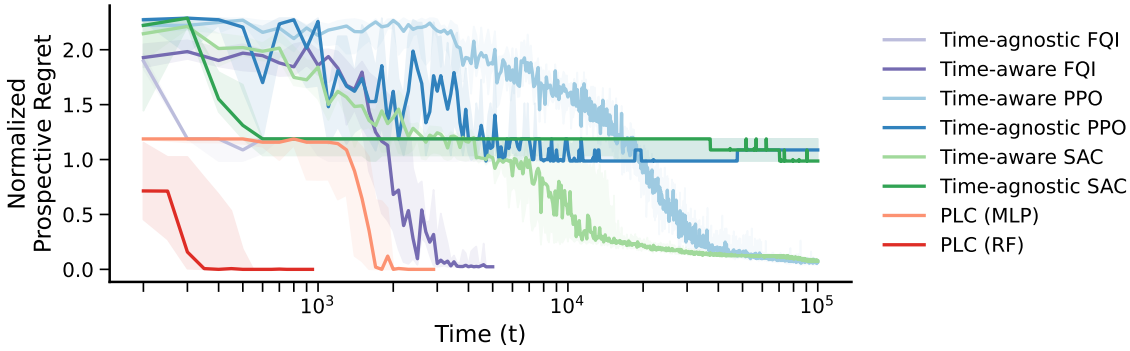


Figure 4.2: Normalized prospective regret as a function of training time for PLC and RL baselines (FQI, SAC, PPO) on the foraging environment. PLC converges to near-zero regret within approximately 100 time steps, while time-aware RL baselines require 10^3 to 10^5 steps — one to three orders of magnitude more experience. Time-agnostic versions of the same RL algorithms plateau at suboptimal regret and never converge. Error bars are computed over 10 random seeds.

Both regressors contribute, but the instantaneous regressor carries most of the signal. An ablation study compares three variants: PLC with both regressors (the full algorithm), PLC-I with only the instantaneous regressor (planning uses Q_{finite} alone, with no terminal cost), and PLC-C with only the cumulative regressor (planning uses Q_{terminal} alone, with no per-step costs). PLC-I performs nearly as well as the full PLC, with both converging at comparable speeds to near-zero regret, as shown in Fig. 4.3. This indicates that the instantaneous regressor, which provides fine-grained per-step loss estimates and serves as the counterfactual imputation mechanism for unvisited states, carries the majority of the planning signal in this environment. PLC-C, by contrast, converges poorly, as the cumulative regressor alone does not provide enough resolution to discriminate between action sequences that differ in their near-term consequences. The full PLC retains a modest advantage over PLC-I in stability and final regret, suggesting that the terminal cost term provides a useful

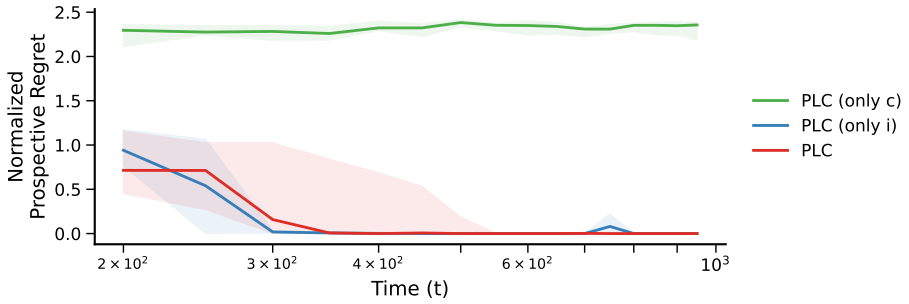


Figure 4.3: Ablation study on the foraging environment. Normalized prospective regret is plotted for the full PLC algorithm (both regressors), PLC-I (instantaneous regressor only), and PLC-C (cumulative regressor only). PLC and PLC-I converge at comparable speeds to near-zero regret, indicating that the instantaneous regressor carries the dominant planning signal. PLC-C converges poorly, confirming that the cumulative regressor alone is insufficient. Error bars are computed over 10 random seeds.

regularizing signal even when the instantaneous regressor is dominant.

4.7 Relationship to reinforcement learning

Prospective learning with control is not reinforcement learning, though it addresses a similar class of problems. The differences are worth stating precisely, both to clarify the contribution and to identify the settings where each framework is more appropriate.

No MDP assumption. Classical RL theory is built on the MDP (Sutton and Barto, 1998): the next state depends only on the current state and action, not on the history. PLC does not require this. The reward function $r_t(s)$ can depend on t in arbitrary ways, and the transition dynamics can depend on the full history. In the foraging task, the reward function is periodic in t, a structure that an MDP cannot represent without augmenting the state space with a time variable. PLC handles this

naturally because time is already an input to both regressors.

No stationarity. Classical RL assumes that the transition and reward functions do not change over time. PLC is designed precisely for settings where they do. The time embedding allows the regressors to track non-stationary rewards, and the planning step evaluates candidate sequences against time-specific predictions.

No resets. Classical RL assumes episodic interaction: the agent experiences the environment in episodes, each starting from a reset to an initial state. PLC operates in a single continuous life. Every action is permanent, and the agent cannot revisit a past state by resetting the environment. This is the setting faced by biological organisms and by deployed systems that cannot be taken offline for retraining.

Estimation vs. dynamic programming. The deepest difference is methodological. RL is rooted in dynamic programming (Bellman, 1954): value functions satisfy the Bellman equation, and algorithms iterate on this recursion. PLC is rooted in estimation theory: the learner fits two regression models and plans by evaluating candidate action sequences against them. The Bellman equation does not appear anywhere in the PLC algorithm. This is not an oversight; it is a consequence of dropping the MDP and stationarity assumptions, which are precisely the conditions under which the Bellman recursion holds.

There is a further connection worth noting. Model-based reinforcement learning (Bertsekas, 2023) also decomposes the problem into learning a model of the environment and planning against it, a structure that resembles PLC's two-regressor-plus-planner

architecture. The key difference is what is learned: model-based RL learns a transition model and a reward model separately and then plans via the Bellman recursion, whereas PLC learns loss predictors (instantaneous and cumulative) and plans by direct sequence evaluation. The two approaches converge, however, on a shared practical question: when the dynamics are unknown, both must learn a forward model, and both suffer from compounding rollout error. The growing literature on world models (W. Zhang et al., 2026), which covers learned simulators of the environment used for planning and imagination, is directly relevant to scaling PLC beyond settings with known dynamics. Integrating a learned world model into the PLC framework, while preserving the time-aware, estimation-theoretic character of the prospective approach, is an open problem at the intersection of prospective learning and model-based RL.

These differences come with tradeoffs. RL's dynamic-programming machinery is extremely powerful when its assumptions hold: it provides efficient algorithms with strong finite-sample guarantees for tabular and linear MDPs. PLC does not yet have comparable finite-sample theory, and establishing explicit convergence rates remains an open problem. Conversely, PLC's estimation-theoretic approach is more flexible: it applies to non-Markov, non-stationary, single-life settings where RL's assumptions break down. The foraging experiment provides one concrete instance where this flexibility translates into a practical advantage, as PLC converges to oracle performance in 100 steps where time-aware RL methods need thousands.

Several recent lines of work in RL are moving toward relaxing the same assumptions that PLC dispenses with. Continual RL (S. Kumar, Marklund, A. Rao, Y. Zhu, Jeon, Liu, et al., 2023) considers agents that learn over extended lifetimes without resets.

Single-life RL (Chen et al., 2022) studies settings where the agent has one chance to act. Non-stationary RL (Wei and Luo, 2021) develops algorithms robust to changing dynamics. PLC can be seen as a framework that starts from the other end, from the estimation-theoretic prospective learning foundation, and arrives at a similar problem space from a different direction. Whether the two lines of work will converge, and what the resulting synthesis will look like, is an open question for future research.

Chapter 5
Synthesis and Outlook

The preceding chapters developed three movements of a single argument. Chapter 2 showed that even the simplest departure from the IID assumption, two fixed distributions with a tunable amount of non-current data, reveals structure that existing theory cannot predict. Chapter 3 introduced a formalism in which time is a first-class variable in learning, proved that ERM over time-indexed hypotheses achieves Bayes risk, and showed that ERM without time provably cannot. Chapter 4 extended the formalism to sequential decision-making, where the learner's actions co-determine the future it must anticipate.

This chapter makes the claim that the three movements are not separate contributions but facets of a single idea: *the right abstraction for learning in a changing world is a stochastic process, the right hypothesis is a function of time, and the right benchmark is a realization-dependent Bayes risk.* Once this abstraction is adopted, the classical learning paradigms, PAC learning, domain adaptation, meta-learning, continual learning, online learning, and reinforcement learning, appear as special cases corresponding to specific assumptions about the temporal structure of the process and the learner's capacity to act. The chapter states this unification explicitly, revisits the natural-intelligence motivation that opened the thesis, and then turns to open problems, both theoretical and practical, that the prospective learning program raises.

5.0.1 A unified time-indexed view

The taxonomy of Section 3.5 organized prospective learning problems by the richness of their temporal structure. This section inverts the perspective: instead of asking "what kinds of problems does prospective learning contain?", it asks "what does each existing paradigm look like from the prospective learning viewpoint?"

PAC learning is prospective learning where the stochastic process is constant: $Z_t \overset{\text{iid}}{\sim} P$ for all t. The prospective Bayes risk R_t^* does not depend on t or on the realization $z_{\leq t}$; it is the classical Bayes risk of P. The hypothesis sequence degenerates to a constant sequence, and prospective ERM reduces to standard ERM. Time carries no information.

Domain adaptation and transfer learning are prospective learning where the index set has two elements. The learner has data from a source distribution P_o and must predict on a target distribution P_t, with at most one shift. The weighted-ERM framework of Chapter 2 is the natural tool. The non-monotonicity result of Section 2.2 shows that even this two-point case has structure that existing theory misses.

Multi-task and meta-learning are prospective learning with a finite task family and no temporal dynamics. Tasks are drawn IID from a meta-distribution, making it impossible to predict which task comes next, so the best the learner can do is prepare for the most likely task. In the prospective framework, this corresponds to a stochastic process whose marginals are exchangeable: the order of the tasks carries no information, and a time-agnostic inductive bias suffices. Prospective learning becomes non-trivially different from meta-learning precisely when the task order is informative,

that is, when there are temporal dynamics to exploit.

Continual learning is prospective learning with a retrospective objective. The continual learner faces a non-stationary stream and seeks to retain performance on past tasks while adapting to new ones. The prospective learner faces the same stream but seeks to *anticipate* future tasks. The two objectives are complementary, as a learner that can anticipate future tasks can also prepare for them, reducing the need for retrospective correction. The experiments of Section 3.6 showed that continual learning baselines, Bayesian gradient descent and online SGD, plateau at chance on problems where prospective ERM converges to Bayes risk. The failure is structural: continual learning methods adapt retrospectively but do not model the dynamics of how tasks evolve.

Online learning is prospective learning without distributional assumptions and with an adversarial benchmark. The online learner minimizes regret, the difference between its cumulative loss and that of the best fixed hypothesis in hindsight. Prospective learning instead minimizes the gap to the Bayes-optimal *sequence* of hypotheses, which is a stronger benchmark when the stochastic process has exploitable structure. Follow-the-Leader, the canonical no-regret algorithm, produces a time-agnostic hypothesis and is therefore subject to the limitations of Proposition 3.3.1.

Reinforcement learning is prospective learning with control (Chapter 4) under the MDP assumption with stationary dynamics and episodic resets. Prospective control relaxes all three assumptions. The MDP assumption is replaced by a general stochastic process; stationarity is replaced by time-indexed reward and transition

models; episodic resets are replaced by single-life operation. The Bellman recursion, which is the foundation of classical RL theory, does not hold without the Markov property, and the PLC algorithm replaces it with an estimation-theoretic approach, using learned loss regressors and sequence-level planning.

The unifying claim is not that prospective learning *replaces* these paradigms, as each has developed powerful tools for its specific setting, but that it provides a single formal framework within which they can be compared, their assumptions made explicit, and their limitations diagnosed. The failure modes documented in this thesis, non-monotonic generalization (Chapter 2), structural incapacity of time-agnostic ERM (Chapter 3 and Proposition 3.3.1), and the inability of retrospective agents to anticipate future rewards (Chapter 4), are all symptoms of the same underlying issue: the learner's formalism does not include time.

5.0.2 Return to natural intelligence

Chapter 1 opened with the observation that natural intelligences are constitutively prospective. Organisms do not merely react to the present; they anticipate the future and act in preparation for it (Kording et al., 2025). The thesis developed a formal framework for this capacity. It is worth asking, now that the formalism is in hand, how well it captures the phenomena that motivated it.

Predictive coding, the hypothesis that neural circuits maintain and update internal models that predict incoming sensory stimuli, corresponds most directly to Scenario 3 of the taxonomy. The brain receives a stream of non-stationary, temporally dependent sensory data and must predict the next input to allocate attention and metabolic

resources efficiently. A prospective learner for Scenario 3, equipped with a model of the temporal dynamics such as a time embedding, does exactly this: it uses the realized past to build a time-indexed predictor that anticipates future inputs. The connection is not merely metaphorical, as the Fourier time embedding used in the experiments of Section 3.6 is a rudimentary version of the oscillatory codes that neural circuits (Klimesch, 2018) use to represent temporal context.

Allostasis, the principle that regulatory systems anticipate the organism's needs and prepare to meet them before they arise, corresponds to Scenario 4. The organism's regulatory actions (e.g., increasing mitochondrial energy production before muscular exertion) influence its future physiological state, and the optimal regulatory policy requires anticipating future demands. This is prospective control: the learner acts, the action affects the future, and the objective is cumulative future well-being. The foraging task of Section 4.6 is a stylized allostatic problem: the agent must leave a depleting resource before it runs out, accepting short-term cost for long-term gain.

Mental time travel, the capacity to project oneself into the future and simulate possible scenarios, corresponds to the hypothesis-sequence view of prospective learning. A prospective learner that outputs a sequence $(h_1, h_2, \ldots)$ is, in effect, committing to a plan that specifies what it will do at each future moment. The planning step in PLC, which enumerates candidate action sequences and scores them against learned models, is a computational analog of mental simulation: the agent imagines possible futures and selects the one with the lowest anticipated cost based on the current internal models.

Prospective memory, the ability to remember to perform intended actions at appropriate future moments, maps onto the time-as-input mechanism. A prospective learner that receives $\varphi(t')$ as input at future time t' can produce a prediction that is specific to that moment, including predictions that are relevant only at that moment (e.g., "leave the patch at $t' = 47$"). The time embedding is, in a sense, a trigger: it tells the learner "it is now time t'; what should you do?" This is precisely the function that prospective memory serves in biological cognition.

What the formalism does not capture. The prospective learning framework, as developed in this thesis, operates under expected-loss objectives. Biological organisms balance expected reward against risk, and their regulatory processes reflect this balance. Allostasis, for instance, errs on the side of over-preparation because the cost of under-preparation can be catastrophic. Extending the prospective risk to incorporate risk-sensitive objectives is a natural direction.

The framework also does not address social prospection (Meulemans et al., 2025), the ability to anticipate the actions and mental states of other agents, or counterfactual reasoning, the ability to reason about what would have happened under different actions. Both are central to human prospective cognition and would require extending the formalism to multi-agent and counterfactual settings.

Finally, the framework does not model meta-cognition about one's own prospective abilities, that is, the capacity to know when one's predictions are likely to be wrong and to seek additional information accordingly. This is related to the time-embedding-selection problem identified in Section 3.6: a learner that knows its time embedding is

mismatched to the temporal structure of the process would benefit from switching to a better one, but the current framework provides no mechanism for this.

5.1 Open problems

The prospective learning program, as developed in this thesis, is at an early stage. The theoretical foundations are in place, the main learnability results are proved, and the empirical evidence is encouraging. But many questions remain open, some technical, some conceptual, and some practical. This section organizes them into four groups.

5.1.1 Theoretical foundations

Weak-vs-strong equivalence. In PAC learning, weak and strong learnability are equivalent in the distribution-agnostic setting (Schapire, 1990). In prospective learning, Proposition 3.3.1 showed that time-agnostic ERM can be weakly but not strongly learnable for specific stochastic processes. Whether weak and strong prospective learnability are equivalent, that is, whether every family of stochastic processes that is weakly learnable is also strongly learnable given a sufficiently rich hypothesis class, remains open.

Complexity measures for stochastic processes. PAC learning has VC dimension; what does prospective learning have? The analogy suggests that there should be a quantity, a prospective complexity, that characterizes the sample complexity of prospective ERM for a family of stochastic processes. Candidates exist in the information-theory literature but have not been connected to the prospective learning framework. Two are particularly promising.

Predictive information, introduced by Bialek et al. (2001) is the mutual information between the past and the future of a stochastic process: $I_{\text{pred}}(t) = I(Z_{\leq t}; Z_{>t})$. It measures how much knowing the past reduces uncertainty about the future, which is exactly the quantity a prospective learner seeks to exploit. Processes with high predictive information have structured futures that can be anticipated; processes with low predictive information are closer to IID and offer little to prospect on. The conjecture, not yet proved, is that prospective sample complexity is inversely related to predictive information: processes with more predictable futures are easier to learn prospectively.

The information bottleneck, proposed by Tishby et al. (2000), provides a complementary perspective. It asks: what is the minimal sufficient representation of the past that preserves information about the future? In the prospective learning context, the time embedding $\varphi(t)$ can be understood as an information bottleneck, compressing the raw time index into a low-dimensional representation that ideally preserves the temporal structure relevant to prediction. The time-embedding-selection problem identified in Section 3.6, choosing φ to match the process, is, from this viewpoint, the problem of finding the right bottleneck. Whether the information-bottleneck framework can guide the choice of φ in a principled way is an open question.

Finite-sample rates. Theorems 3.4.1 and 3.4.2 are asymptotic: they guarantee convergence but do not specify rates. The periodic case in Section 3.4 gives explicit sample complexity, but the general case, particularly for non-periodic and non-Markov processes, remains open. For prospective control (Chapter 4), the situation is even less developed: Theorem 4.3.1 guarantees asymptotic Bayes optimality but says nothing

about how many time steps are needed.

Learnability beyond finite families. Both Theorems 3.4.1 and 3.4.2 assume a finite family $\mathcal{Z}$ of stochastic processes. Extending the results to infinite families, parameterized by continuous quantities such as the period of a periodic process or the drift rate of a linear process, would significantly broaden the theory's applicability. The challenge is that the uniform-concentration condition (Theorem 3.4.1, condition (ii)) must hold over the family, and uniformity over an infinite family is a stronger requirement.

5.1.2 Connections to control theory and dynamical systems

Adaptive control. The prospective control framework of Chapter 4 has deep connections to adaptive control (Recht, 2019; Dean et al., 2018), the branch of control theory concerned with controllers that adjust their parameters in real time as the system's dynamics change. Classical adaptive control methods, such as model reference adaptive control and self-tuning regulators, address non-stationary systems by estimating the plant's parameters online and adjusting the control law accordingly. PLC does something similar, estimating reward models online and planning against them, but from an estimation-theoretic rather than control-theoretic starting point. The key difference is that adaptive control typically assumes a parametric model of the dynamics (e.g., a linear system with unknown coefficients), whereas PLC uses regressors and makes no structural assumption about the dynamics beyond what the time embedding can capture.

Model predictive control (MPC). The PLC algorithm's planning step, which enumerates candidate action sequences, scores them against learned models, and selects the best, is structurally identical to model predictive control (Bertsekas, 2023), the workhorse of modern control engineering. MPC solves a finite-horizon optimization problem at each time step using a model of the system dynamics, executes the first action, and re-plans at the next step. PLC does the same, with two differences: the "model" is a pair of learned regressors rather than a physics-based simulator, and the regressors are time-indexed, allowing them to capture non-stationary dynamics. The connection suggests that PLC could benefit from the extensive MPC literature on constraint handling, robustness, and computational efficiency. Conversely, the prospective learning viewpoint could enrich MPC by providing a formal framework for learning the models that MPC plans against, with learnability guarantees.

World models. A recurring theme in both model-based reinforcement learning (Hafner, Lillicrap, Ba, et al., 2019; Hafner, Lillicrap, Fischer, et al., 2019; D. Ha and Schmidhuber, 2018) and model predictive control is the use of a learned simulator, a *world model*, that allows the agent to imagine the consequences of candidate actions without executing them in the real environment. The PLC algorithm, as noted in Section 4.4, currently assumes known transition dynamics; relaxing this assumption requires learning a forward model $\hat{f}(s_t, a_t, t) \approx s_{t+1}$ alongside the loss regressors. The prospective learning viewpoint adds a specific requirement that generic world models do not address: the forward model must itself be time-indexed, because the dynamics it simulates may be non-stationary. A world model trained on data from times 1 through t must be able to predict state transitions at future times $t' > t$, even if the dynamics

at t' differ from those observed so far. This is a prospective learning problem in its own right, as the world model is a prospective learner whose label is the next state rather than a class label. The compounding-error problem that plagues multi-step rollouts in model-based RL is likely to be exacerbated by non-stationarity, since model errors at early rollout steps shift the predicted state trajectory into regions of the state-time space where the model has little training data. Whether the time-indexed world models can be trained efficiently enough to support real-time planning is an open question at the intersection of prospective learning, model-based RL, and control theory.

Computational mechanics and ϵ-machines. Shalizi and Crutchfield (2001) introduced the concept of an ϵ-machine, a minimal sufficient statistic for predicting the future of a stochastic process from its past. An ϵ-machine's partitions the set of all possible pasts into equivalence classes (causal states) such that two pasts are equivalent if and only if they induce the same conditional distribution over futures. This is precisely the information a prospective learner needs: a representation of the past that preserves everything relevant to the future and discards everything else.

The connection to prospective learning is suggestive but not yet formal. The time embedding $\varphi(t)$, combined with the learned hypothesis $h(\varphi(t), x)$, can be understood as an approximation to the ϵ-machines's causal-state representation. The hypothesis implicitly partitions past-and-time pairs into equivalence classes (regions of the input-time space that receive the same prediction). Whether this partition converges to the ϵ-machine's causal states, and whether the ϵ-machine's structural complexity, that is the statistical complexity C_μ, governs the sample complexity of prospective

learning, are open questions that could connect prospective learning to the rich theory of computational mechanics.

5.1.3 Algorithmic and architectural questions

The time-embedding-selection problem. The experiments of Section 3.6 showed that the time embedding φ must match the temporal structure of the stochastic process: Fourier for periodic, monomial for polynomial drift, and mismatches cause failure. When the temporal structure is unknown, as it will be in most practical applications, how should φ be chosen?

Several approaches are conceivable. The learner could maintain a library of candidate embeddings and select among them by cross-validation on the prospective empirical risk. It could use a learned embedding, a small neural network that maps raw time to a representation trained end-to-end with the main hypothesis, at the cost of additional parameters and potential overfitting. Or it could use a universal embedding that is rich enough to represent any temporal structure, if such an embedding exists. The Fourier embedding with a sufficiently large number of frequencies is universal in the sense of approximation theory, but may require impractically many dimensions for processes with complex temporal structure. Whether there exists a compact, learnable, universal time embedding is an open question with both theoretical and practical significance.

Scaling PLC to complex environments. The PLC algorithm's planning step enumerates all feasible action sequences of length H, which is tractable only for small action spaces and short horizons. For environments with large or continuous action

spaces, approximate planning methods, such as Monte Carlo tree search (Coulom, 2007), learned policy networks, cross-entropy method (CEM) (Pinneri et al., 2021), or sampling-based optimization Nicolò Cesa-Bianchi, Gentile, et al., 2017, will be necessary. The interaction between approximate planning and the two-regressor architecture has not been studied.

Learning a world model. As noted in Section 4.4, PLC currently assumes known transition dynamics. Relaxing this assumption requires learning a forward model $\hat{f}(s_t, a_t, t) \approx s_{t+1}$ alongside the loss regressors. The compounding-error problem, where small errors in $\hat{f}$ accumulate over multi-step rollouts, is well-studied in model-based RL but has not been analyzed in the prospective learning context, where the model must also be time-indexed to capture non-stationary dynamics.

5.1.4 Empirical frontiers

Continuous state, action, and time. All experiments in this thesis use discrete state and action spaces and discrete time. Extending to continuous domains, including continuous state spaces such as robotic manipulation, continuous action spaces such as torque control, and continuous time such as irregularly sampled physiological data, is necessary for real-world deployment. The theoretical framework accommodates continuous time by replacing sums with integrals, but the algorithmic and architectural implications have not been explored.

Stochastic dynamics. The foraging environment of Chapter 4 has deterministic dynamics: the agent's next state is a deterministic function of its current state and action. Real environments are stochastic, and PLC's planning step must be modified to

handle uncertainty in the state trajectory. This connects to the world-model question above, as a stochastic forward model would produce a distribution over future states rather than a single trajectory, and the planning step would need to optimize expected cost over this distribution.

Real-world applications. The prospective learning framework is motivated by real-world problems, including deployed ML systems that degrade as data drifts, physiological monitoring systems that must adapt to changing user states, and autonomous agents that must anticipate environmental changes. Demonstrating that prospective learning provides practical benefits in these settings — beyond the synthetic, MNIST, and CIFAR-10 experiments of this thesis — is the ultimate test of the framework's utility.

Foundation models and prospective pretraining. An intriguing open question is whether a foundation model could be pretrained with a prospective objective, trained not just to predict the next token but to predict tokens at specified future times, with time as an explicit input. Such a model would be, in effect, a general-purpose prospective learner. Whether this is feasible, and whether it would improve performance on downstream tasks with non-stationary data, is unknown.

5.2 Concluding remarks

This thesis began with a simple observation: all learning is for the future, but most of learning theory ignores time. The observation is old, implicit in the work of Seligman and colleagues on prospection in natural intelligence, and in decades of work on

non-stationary learning, continual adaptation, and sequential decision-making. What is new here is the formalization: a definition of prospective learnability, a theorem that identifies the conditions under which it is achievable, an algorithm that achieves it in practice, and an extension to settings where the learner acts on its environment.

The central message is that time is cheap. Making time an input to a hypothesis, by concatenating a time embedding to a neural network's features or appending it to the columns of a decision tree's input matrix, is a trivial modification to any learning system. But it changes what the system can do. A time-agnostic learner can adapt to the present; a time-aware learner can anticipate the future. The gap between the two, as Proposition 3.3.1 and the experiments of this thesis show, is not marginal — it is the difference between converging to Bayes risk and plateauing at chance.

Much remains to be done. The theory needs sharper sample-complexity bounds, a complexity measure for stochastic processes, and extensions to infinite families and continuous domains. The algorithms need world models, scalable planning, and principled time-embedding selection. The framework needs real-world validation, specifically deployed systems that demonstrate the practical benefits of prospection. But the foundation is in place: a formal framework that makes time a first-class variable in learning, and that brings statistical learning one step closer to the prospective character of natural intelligence.

Bibliographic references

Arjovsky, Martin, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz (2019). "Invariant risk minimization." In: *arXiv preprint arXiv:1907.02893*.

Bai, Yuxin, Aranyak Acharyya, Ashwin De Silva, Zeyu Shen, James Hassett, and Joshua T Vogelstein (2025). "Optimal control of the future via prospective learning with control." In: *8th Annual Learning for Dynamics & Control Conference*.

Bai, Yuxin, Cecelia Shuai, Ashwin De Silva, Siyu Yu, Pratik Chaudhari, and Joshua T Vogelstein (2026). "Prospective learning in retrospect." en. In: *Lecture Notes in Computer Science*. Lecture notes in computer science. Cham: Springer Nature Switzerland, pp. 17–29.

Baxter, Jonathan (2000). "A model of inductive bias learning." In: *J. Artif. Intell. Res.* 12.1, pp. 149–198.

Bellman, Richard (1954). "The theory of dynamic programming." In: *Bulletin of the American Mathematical Society* 60.6, pp. 503–515.

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan (2010). "A theory of learning from different domains." In: *Machine learning* 79.1, pp. 151–175.

Ben-David, Shai and Ruth Urner (2012). "On the hardness of domain adaptation and the utility of unlabeled target samples." In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 139–153.

Bertsekas, Dimitri (2023). *A course in Reinforcement Learning*. en. Athena Scientific. 421 pp.

Bialek, William, Ilya Nemenman, and Naftali Tishby (2001). "Predictability, complexity, and learning." In: *Neural computation* 13.11, pp. 2409–2463.

Cesa-Bianchi, Nicolo and Gábor Lugosi (2006). *Prediction, learning, and games*. Cambridge university press.

Cesa-Bianchi, Nicolò, Claudio Gentile, Gábor Lugosi, and Gergely Neu (2017). "Boltzmann exploration done right." In: *Advances in neural information processing systems* 30.

Cesa-Bianchi, Nicolò and Francesco Orabona (2021). "Online learning algorithms." In: *Annual review of statistics and its application* 8.1, pp. 165–190.

Charnov, Eric L (1976). "Optimal foraging, the marginal value theorem." In: *Theoretical population biology* 9.2, pp. 129–136.

Chen, Annie, Archit Sharma, Sergey Levine, and Chelsea Finn (2022). "You only live once: single-life reinforcement learning." In: *Advances in Neural Information Processing Systems* 35, pp. 14784–14797.

Cortes, Corinna, Mehryar Mohri, and Andrés Munoz Medina (2019). "Adaptation based on generalized discrepancy." In: *The Journal of Machine Learning Research* 20.1, pp. 1–30.

Costa, Vincent D, Valery L Tran, Janita Turchi, and Bruno B Averbeck (2015). "Reversal learning and dopamine: a bayesian perspective." In: *Journal of Neuroscience* 35.6, pp. 2407–2416.

Coulom, Rémi (2007). "Efficient selectivity and backup operators in monte-carlo tree search." In: *Computers and Games, CG 2006, Turin, Italy, May 29–31, 2006, Revised Papers*. Ed. by H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. Donkers. Lecture Notes in Computer Science. Springer, pp. 72–83.

Darlow, Luke N, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey (2018). "Cinic-10 is not imagenet or cifar-10." In: *arXiv preprint arXiv:1810.03505*.

De Lange, Matthias and Tinne Tuytelaars (2021). "Continual prototype evolution: learning online from non-stationary data streams." In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8250–8259.

De Silva, Ashwin, Rahul Ramesh, Carey Priebe, Pratik Chaudhari, and Joshua T Vogelstein (2023a). "The value of out-of-distribution data." In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. proceedings.mlr.press, pp. 7366–7389.

De Silva, Ashwin, Rahul Ramesh, Lyle Ungar, Marshall Hussain Shuler, Noah J Cowan, Michael Platt, Chen Li, Leyla Isik, Seung-Eon Roh, Adam Charles, Archana Venkataraman, Brian Caffo, Javier J How, Justus M Kebschull, John W Krakauer, Maxim Bichuch, Kaleab Alemayehu Kinfu, Eva Yezerets, Dinesh Jayaraman, Jong M Shin, Soledad Villar, Ian Phillips, Carey E Priebe, Thomas Hartung, Michael I Miller, Jayanta Dey, Ningyuan Huang, Eric Eaton, Ralph Etienne-Cummings, Elizabeth L Ogburn, Randal Burns, Onyema Osuagwu, Brett Mensh, Alysson R Muotri, Julia Brown, Chris White, Weiwei Yang, Andrei A Rusu Timothy Verstynen, Konrad P Kording, Pratik Chaudhari, and Joshua T Vogelstein (2023b). "Prospective Learning: Principled Extrapolation to the Future." In: *Proceedings of The 2nd Conference on Lifelong Learning Agents*. Ed. by Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup. Vol. 232. Proceedings of Machine Learning Research. PMLR, pp. 347–357.

Dean, Sarah, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu (2018). "Regret bounds for robust adaptive control of the linear quadratic regulator." In: *Advances in Neural Information Processing Systems* 31.

Devroye, Luc, Laszlo Györfi, and Gabor Lugosi (1997). *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. Corrected edition.

Ernst, Damien, Pierre Geurts, and Louis Wehenkel (2005). "Tree-based batch mode reinforcement learning." In: *Journal of Machine Learning Research* 6.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks." In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 1126–1135.

Finn, Chelsea, Aravind Rajeswaran, Sham Kakade, and Sergey Levine (2019). "Online Meta-Learning." In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 1920–1930.

Friston, Karl and Stefan Kiebel (2009). "Predictive coding under the free-energy principle." In: *Philosophical transactions of the Royal Society B: Biological sciences* 364.1521, pp. 1211–1221.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky (2016). "Domain-adversarial training of neural networks." In: *The journal of machine learning research* 17.1, pp. 2096–2030.

Ghifary, Muhammad, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi (2015). "Domain generalization for object recognition with multi-task autoencoders." In: *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559.

Ghosh, B K and P K Sen (1991). *Handbook of Sequential Analysis (Statistics: A Series of Textbooks and Monographs)*. en. 1st ed. CRC Press.

Glivenko, V (1933). "Sulla determinazione empirica delle leggi di probabilita." In: *Gion. Ist. Ital. Attauri.* 4, pp. 92–99.

Glivenko, Valery (1933). "Sulla determinazione empirica delle leggi di probabilita." In: *Gion. Ist. Ital. Attauri.* 4, pp. 92–99.

Gneiting, Tilmann and Matthias Katzfuss (2014). "Probabilistic forecasting." en. In: *Annu. Rev. Stat. Appl.* 1.1, pp. 125–151.

Gulrajani, Ishaan and David Lopez-Paz (2020). "In search of lost domain generalization." In: *arXiv preprint arXiv:2007.01434*.

Guo, Daya, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. (2025). "Deepseek-r1: incentivizing reasoning capability in llms via reinforcement learning." In: *arXiv preprint arXiv:2501.12948*.

Ha, David and Jürgen Schmidhuber (2018). "World models." In: *arXiv preprint arXiv:1803.10122* 2.3, p. 440.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine (2018). "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor." In: *International conference on machine learning*. Pmlr, pp. 1861–1870.

Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. (2018). "Soft actor-critic algorithms and applications." In: *arXiv preprint arXiv:1812.05905*.

Hafner, Danijar, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi (2019). "Dream to control: learning behaviors by latent imagination." In: *arXiv preprint arXiv:1912.01603*.

Hafner, Danijar, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson (2019). "Learning latent dynamics for planning from pixels." In: *International conference on machine learning*. PMLR, pp. 2555–2565.

Hanneke, Steve (2021). "Learning whenever learning is possible: universal learning under general stochastic processes." In: *J. Mach. Learn. Res.* 22, 130:1–130:116.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn." In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Helm, Hayden, Ashwin De Silva, Joshua T Vogelstein, Carey E Priebe, and Weiwei Yang (2024). "Approximately optimal domain adaptation with fisher's linear discriminant." In: *Mathematics* 12.5, p. 746.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models." In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Huang, Yanping and Rajesh P N Rao (2011). "Predictive coding." In: *Wiley interdisciplinary reviews. Cognitive science* 2.5, pp. 580–593.

Izquierdo, Alicia, Jonathan L Brigman, Anna K Radke, Peter H Rudebeck, and Andrew Holmes (2017). "The neural basis of reversal learning: an updated perspective." In: *Neuroscience* 345, pp. 12–26.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. (2021). "Highly accurate protein structure prediction with alphafold." In: *nature* 596.7873, pp. 583–589.

Khetarpal, Khimya, Matthew Riemer, Irina Rish, and Doina Precup (2022). "Towards continual reinforcement learning: a review and perspectives." In: *Journal of Artificial Intelligence Research* 75, pp. 1401–1476.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. (2017). "Overcoming catastrophic forgetting in neural networks." In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Klimesch, Wolfgang (2018). "The frequency architecture of brain and brain body oscillations: an analysis." In: *European Journal of Neuroscience* 48.7, pp. 2431–2453.

Koh, Pang Wei, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. (2021). "Wilds: a benchmark of in-the-wild distribution shifts." In: *International Conference on Machine Learning*. PMLR, pp. 5637–5664.

Kording, Konrad P, Joshua T Vogelstein, Pratik Chaudhari, and Timothy Verstynen (2025). "Toward a science of prospective learning." In: *Neuron* 113.24, pp. 4103–4106.

Krizhevsky, Alex (2009). *Learning Multiple Layers of Features from Tiny Images*. Tech. rep.

Kumar, Saurabh, Henrik Marklund, Ashish Rao, Yifan Zhu, Hong Jun Jeon, Yueyang Liu, and Benjamin Van Roy (2023). "Continual Learning as Computationally Constrained Reinforcement Learning." In: arXiv: 2307.04345 [cs.LG].

Kumar, Saurabh, Henrik Marklund, Ashish Rao, Yifan Zhu, Hong Jun Jeon, Liu Yueyang, and Benjamin Van Roy (2025). "Continual learning as computationally constrained reinforcement learning." In: *Foundations and Trends® in Machine Learning* 18.4, pp. 913–1053.

Levine, Sergey, Aviral Kumar, George Tucker, and Justin Fu (2020). "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." In: *arXiv [cs.LG]*. arXiv: 2005.01643 [cs.LG].

Li, Da, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales (2017a). "Deeper, broader and artier domain generalization." In: *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550.

— (2017b). *Deeper, Broader and Artier Domain Generalization*. arXiv: 1710.03077 [cs.CV].

Lopez-Paz, David and Marc'Aurelio Ranzato (2017). "Gradient episodic memory for continual learning." In: *Advances in neural information processing systems* 30.

Mansour, Yishay, Mehryar Mohri, and Afshin Rostamizadeh (2008). "Domain adaptation with multiple sources." In: *Advances in neural information processing systems* 21.

Maurer, Andreas and Tommi Jaakkola (2005). "Algorithmic stability and meta-learning." In: *Journal of Machine Learning Research* 6.6.

McDaniel, Mark A and Gilles O Einstein (2007). "Prospective memory: an overview and synthesis of an emerging field." In.

Meulemans, Alexander, Rajai Nasser, Maciej Wolczyk, Marissa A Weis, Seijin Kobayashi, Blake Richards, Guillaume Lajoie, Angelika Steger, Marcus Hutter, James Manyika, et al. (2025). "Embedded universal predictive intelligence: a coherent framework for multi-agent learning." In: *arXiv preprint arXiv:2511.22226*.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781*.

Mohri, Mehryar and Andres Muñoz Medina (2012). "New analysis and algorithm for learning with drifting distributions." In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 124–138.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of machine learning*. MIT press.

Munos, Remi and Csaba Szepesvari (2008). "Finite-time bounds for fitted value iteration." In: *Journal of Machine Learning Research* 9.5.

Pan, Sinno Jialin, Ivor W Tsang, James T Kwok, and Qiang Yang (2010). "Domain adaptation via transfer component analysis." In: *IEEE transactions on neural networks* 22.2, pp. 199–210.

Peng, Xingchao, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang (2019). "Moment matching for multi-source domain adaptation." In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415.

Petropoulos, Fotios, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J Bessa, Jakub Bijak, John E Boylan, Jethro Browell, Claudio Carnevale, Jennifer L Castle, Pasquale Cirillo, Michael P Clements, Clara Cordeiro, Fernando Luiz Cyrino Oliveira, Shari De Baets, Alexander Dokumentov, Joanne Ellison, Piotr Fiszeder, Philip Hans Franses, David T Frazier, Michael Gilliland, M Sinan Gönül, Paul Goodwin, Luigi Grossi, Yael Grushka-Cockayne, Mariangela Guidolin, Massimo Guidolin, Ulrich Gunter, Xiaojia Guo, Renato Guseo, Nigel Harvey, David F Hendry, Ross Hollyman, Tim Januschowski, Jooyoung Jeon, Victor Richmond R Jose, Yanfei Kang, Anne B Koehler, Stephan Kolassa, Nikolaos Kourentzes, Sonia Leva, Feng Li, Konstantia Litsiou, Spyros Makridakis, Gael M Martin, Andrew B Martinez, Sheik Meeran, Theodore Modis, Konstantinos Nikolopoulos, Dilek Önkal, Alessia Paccagnini, Anastasios Panagiotelis, Ioannis Panapakidis, Jose M Pavía, Manuela Pedio, Diego J Pedregal, Pierre Pinson, Patrícia Ramos, David E Rapach, J James Reade, Bahman Rostami-Tabar, Michal Rubaszek, Georgios Sermpinis, Han Lin Shang, Evangelos Spiliotis, Aris A Syntetos, Priyanga Dilini Talagala, Thiyanga S Talagala, Len Tashman, Dimitrios Thomakos, Thordis Thorarinsdottir, Ezio Todini, Juan Ramón Trapero Arenas, Xiaoqian Wang, Robert L Winkler, Alisa Yusupova, and Florian Ziel (2022). "Forecasting: theory and practice." In: *International journal of forecasting* 38.3, pp. 705–871. DOI: 10.1016/j.ijforecast.2021.11.001.

Pinneri, Cristina, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius (2021). "Sample-efficient cross-entropy method for real-time planning." In: *Conference on Robot Learning*. PMLR, pp. 1049–1065.

Pyke, Graham H (1984). "Optimal foraging theory: a critical review." In: *Annual review of ecology and systematics* 15, pp. 523–575.

Quinonero-Candela, Joaquin, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence (2008). *Dataset shift in machine learning*. Mit Press.

Raby, C R and N S Clayton (2009). "Prospective cognition in animals." en. In: *Behav. Processes* 80.3, pp. 314–324.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. (2021). "Learning transferable visual models from natural language supervision." In: *International conference on machine learning*. PmLR, pp. 8748–8763.

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever (2023). "Robust speech recognition via large-scale weak supervision." In: *International conference on machine learning*. PMLR, pp. 28492–28518.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. (2018). "Improving language understanding by generative pre-training." In.

Ramesh, Rahul and Pratik Chaudhari (2021). "Model zoo: a growing" brain" that learns continually." In: *arXiv preprint arXiv:2106.03027*.

Recht, Benjamin (2019). "A tour of reinforcement learning: the view from continuous control." In: *Annual Review of Control, Robotics, and Autonomous Systems* 2.1, pp. 253–279.

Reddi, Sashank, Barnabas Poczos, and Alex Smola (2015). "Doubly robust covariate shift correction." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1.

Romera-Paredes, Bernardino and Philip Torr (2015). "An embarrassingly simple approach to zero-shot learning." en. In: *International Conference on Machine Learning*. PMLR, pp. 2152–2161.

Schapire, Robert E (1990). "The strength of weak learnability." In: *Machine learning* 5, pp. 197–227.

Schneider, Steffen, Alexei Baevski, Ronan Collobert, and Michael Auli (2019). "Wav2vec: unsupervised pre-training for speech recognition." In: *arXiv preprint arXiv:1904.05862*.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal policy optimization algorithms." In: *arXiv preprint arXiv:1707.06347*.

Seligman, Martin E P, Peter Railton, Roy F Baumeister, and Chandra Sripada (2016). *Homo Prospectus*. en. Vol. 384. New York, NY, US: Oxford University Press.

— (2013). "Navigating into the future or driven by the past." In: *Perspectives on psychological science* 8.2, pp. 119–141.

Senior, Andrew W, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. (2020). "Improved protein structure prediction using potentials from deep learning." In: *Nature* 577.7792, pp. 706–710.

Shalev-Shwartz, Shai (2025). "Online learning and online convex optimization." In: *Foundations and Trends® in Machine Learning* 4.2, pp. 107–194.

Shalizi, Cosma Rohilla and James P Crutchfield (2001). "Computational Mechanics: Pattern and Prediction, Structure and Simplicity." In: *Journal of statistical physics* 104.3, pp. 817–879. DOI: 10.1023/A:1010388907793.

Silva, Ashwin De, Rahul Ramesh, Rubing Yang, Siyu Yu, Joshua T Vogelstein, and Pratik Chaudhari (2024). "Prospective learning: learning for a dynamic future." In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Snell, Jake, Kevin Swersky, and Richard Zemel (2017). "Prototypical networks for few-shot learning." In: *Advances in neural information processing systems* 30.

Song, Yuhang, Beren Millidge, Tommaso Salvatori, Thomas Lukasiewicz, Zhenghua Xu, and Rafal Bogacz (2024). "Inferring neural activity before plasticity as a foundation for learning beyond backpropagation." In: *Nature neuroscience* 27.2, pp. 348–358.

Sterling, Peter (2012). "Allostasis: a model of predictive regulation." en. In: *Physiology & behavior* 106.1, pp. 5–15. DOI: 10.1016/j.physbeh.2011.06.004.

Strehl, Alexander L, Lihong Li, and Michael L Littman (2009). "Reinforcement learning in finite MDPs: PAC analysis." In: *Journal of machine learning research: JMLR* 10.84, pp. 2413–2444.

Suddendorf, Thomas, Donna Rose Addis, and Michael C Corballis (2009). "Mental time travel and the shaping of the human mind." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1521, pp. 1317–1324.

Sugiyama, Masashi, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul Von Bünau, and Motoaki Kawanabe (2008). "Direct importance estimation for covariate shift adaptation." In: *Annals of the Institute of Statistical Mathematics* 60.4, pp. 699–746.

Sun, Baochen and Kate Saenko (2016). "Deep coral: correlation alignment for deep domain adaptation." In: *European conference on computer vision*. Springer, pp. 443–450.

Sutton, Richard S and Andrew G Barto (1998). *Introduction to Reinforcement Learning*. Camgridge: MIT Press.

Thrun, Sebastian (1998). "Lifelong learning algorithms." In: *Learning to learn*. Springer, pp. 181–209.

Tishby, Naftali, Fernando C Pereira, and William Bialek (2000). "The information bottleneck method." In: *arXiv preprint physics/0004057*.

Valiant, L G (1984). "A theory of the learnable." In: *Commun. ACM* 27.11, pp. 1134–1142.

Valiant, Leslie (2013). *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. en. Basic Books.

Van de Ven, Gido M and Andreas S Tolias (2019). "Three scenarios for continual learning." In: *arXiv preprint arXiv:1904.07734*.

Vapnik, Vladimir (1991). "Principles of risk minimization for learning theory." In: *Advances in neural information processing systems* 4.

— (1998). *Statistical Learning Theory*. John Wiley & Sons.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in Neural Information Processing Systems*. Vol. 30.

Vogelstein, Joshua T, Jayanta Dey, Hayden S Helm, Will LeVine, Ronak D Mehta, Tyler M Tomita, Haoyin Xu, Ali Geisa, Qingyang Wang, Gido M van de Ven, Chenyu Gao, Weiwei Yang, Bryan Tower, Jonathan Larson, Christopher M White, and Carey E Priebe (2020). "A simple lifelong learning approach." In: *arXiv [cs.AI]*.

Vogelstein, Joshua T, Timothy Verstynen, Konrad P Kording, Leyla Isik, John W Krakauer, Ralph Etienne-Cummings, Elizabeth L Ogburn, Carey E Priebe, Randal Burns, Kwame Kutten, James J Knierim, James B Potash, Thomas Hartung, Lena Smirnova, Paul Worley, Alena Savonenko, Ian Phillips, Michael I Miller, Rene Vidal, Jeremias Sulam, Adam Charles, Noah J Cowan, Maxim Bichuch, Archana Venkataraman, Chen Li, Nitish Thakor, Justus M Kebschull, Marilyn Albert, Jinchong Xu, Marshall Hussain Shuler, Brian Caffo, Tilak Ratnanather, Ali Geisa, Seung-Eon Roh, Eva Yezerets, Meghana Madhyastha, Javier J How, Tyler M Tomita, Jayanta Dey, Ningyuan, Huang, Jong M Shin, Kaleab Alemayehu Kinfu, Pratik Chaudhari, Ben Baker, Anna Schapiro, Dinesh Jayaraman, Eric Eaton, Michael Platt, Lyle Ungar, Leila Wehbe, Adam Kepecs, Amy Christensen, Onyema Osuagwu, Bing Brunton, Brett Mensh, Alysson R Muotri, Gabriel Silva, Francesca Puppo, Florian Engert, Elizabeth Hillman, Julia Brown, Chris White, and Weiwei Yang (2022). "Prospective Learning: Back to the Future." In: *arXiv preprint arXiv:2201.07372*. arXiv: 2201.07372 [cs.LG].

Wei, Chen-Yu and Haipeng Luo (2021). "Non-stationary reinforcement learning without prior knowledge: an optimal black-box approach." In: *Conference on learning theory*. PMLR, pp. 4300–4354.

Yao, Huaxiu, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn (2022). "Wild-time: a benchmark of in-the-wild distribution shift over time." In: *Advances in Neural Information Processing Systems* 35, pp. 10309–10324.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide residual networks." In: *arXiv preprint arXiv:1605.07146*.

Zenke, Friedemann, Ben Poole, and Surya Ganguli (2017). "Continual Learning Through Synaptic Intelligence." In: *International Conference on Machine Learning*, pp. 3987–3995.

Zeno, Chen, Itay Golan, Elad Hoffer, and Daniel Soudry (2021). "Task-agnostic continual learning using online variational bayes with fixed-point updates." In: *Neural Computation* 33.11, pp. 3139–3177.

Zhang, Wancong, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, et al. (2026). "Hierarchical planning with latent world models." In: *arXiv preprint arXiv:2604.03208*.

Žliobaitė, Indrė, Mykola Pechenizkiy, and Joao Gama (2015). "An overview of concept drift applications." In: *Big data analysis: new algorithms for a new society*, pp. 91–114.

Appendix A

Supplementary materials for Chapter 2

A.1 Fisher's Linear Discriminant

In this section, we derive Fisher's Linear Discriminant (FLD) decision rule when we have samples from a single distribution. Suppose we have a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ drawn from a distribution P on $\mathcal{X} \times \{0, 1\}$.

Let f_k and π_k be the class conditional density and prior probability of class k ($k \in \{0, 1\}$) respectively. The probability that x belongs to class k is

$$\mathbb{P}\left[Y = k \mid X = x\right] = \frac{\pi_k f_k(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)},$$

and the maximum *a posteriori* estimate of the class label is

$$h(x) = \underset{k \in \{0,1\}}{\operatorname{argmax}} \ \mathbb{P}\left[Y = k \mid X = x\right] = \underset{k \in \{0,1\}}{\operatorname{argmax}} \ \log(\pi_k f_k(x)). \tag{A.1}$$

FLD assumes that each f_k is a multivariate Gaussian distribution with the same covariance matrix Σ, i.e,

$$f_k(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^\top \Sigma^{-1}(x - \mu_k) \right).$$

Under this assumption, the joint-density f of (x, y) becomes,

$$f(x,y) \propto \prod_{k=0}^{1} \left[\frac{\pi_k}{|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k) \right) \right]^{\mathbb{1}[y=k]}$$

Therefore, the log-likelihood $l(\mu_0, \mu_1, \Sigma, \pi_0, \pi_1)$ over $\mathcal{D}$ is given by,

$$\ell(\mu_0, \mu_1, \Sigma, \pi_0, \pi_1) = \sum_{k=0}^{1} \sum_{(x,y)\in\mathcal{D}} \left[\log \pi_k - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k) \right] + \text{constant}$$

where $\mathcal{D}_k$ is the set of samples of $\mathcal{D}$ that belongs to class k. Based on the likelihood function above, we can obtain the maximum likelihood estimates $\hat{\mu}_k, \hat{\Sigma}, \hat{\pi}_k$. The expression for the estimate $\hat{\mu}_k$ is

$$\hat{\mu}_k = \frac{1}{|D_{t,k}|} \sum_{(x,y)\in D_{t,k}} x. \tag{A.2}$$

Plugging these estimates into Eq. (A.1), we get,

$$\begin{aligned}
\hat{h}(x) &= \arg\max_{k\in\{0,1\}} \left[\log \hat{\pi}_k - \frac{1}{2}\log|\hat{\Sigma}| - \frac{1}{2}(x-\hat{\mu}_k)^\top \hat{\Sigma}^{-1}(x-\hat{\mu}_k) \right] \\
&= \arg\max_{k\in\{0,1\}} \left[\log \hat{\pi}_k - \frac{1}{2}\log|\hat{\Sigma}| + x^\top \hat{\Sigma}^{-1}\hat{\mu}_k - \frac{1}{2}\hat{\mu}_k^\top \hat{\Sigma}^{-1}\mu_k \right]
\end{aligned}$$

Therefore, $\hat{h}(x) = 1$ iff,

$$\begin{aligned}
x^\top \hat{\Sigma}^{-1}\hat{\mu}_1 - \frac{1}{2}\hat{\mu}_1^\top \hat{\Sigma}^{-1}\mu_1 + \log\hat{\pi}_1 &> x^\top \hat{\Sigma}^{-1}\hat{\mu}_0 - \frac{1}{2}\hat{\mu}_0^\top \hat{\Sigma}^{-1}\mu_0 + \log\hat{\pi}_0 \\
x^\top \hat{\Sigma}^{-1}\hat{\mu}_1 - x^\top \hat{\Sigma}^{-1}\hat{\mu}_0 &> \frac{1}{2}\hat{\mu}_1^\top \hat{\Sigma}^{-1}\mu_1 - \frac{1}{2}\hat{\mu}_0^\top \hat{\Sigma}^{-1}\mu_0 + \log\hat{\pi}_0 - \log\hat{\pi}_1 \\
(\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0))^\top x &> (\hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0))^\top \left(\frac{\hat{\mu}_0 + \hat{\mu}_1}{2} \right) + \log\frac{\hat{\pi}_0}{\hat{\pi}_1}
\end{aligned}$$

Hence the FLD decision rule $\hat{h}(x)$ is

$$\hat{h}(x) = \begin{cases} 1, & \omega^\top x > c \\ 0, & \text{otherwise} \end{cases}$$

where $\omega = \hat{\Sigma}^{-1}(\hat{\mu}_1 - \hat{\mu}_0)$ is a projection vector and $c = \omega^\top\left(\frac{\hat{\mu}_0+\hat{\mu}_1}{2}\right) + \log\frac{\hat{\pi}_0}{\hat{\pi}_1}$ is a threshold. When $d = 1$ and $\pi_0 = \pi_1$, the decision rule reduces to

$$\hat{h}(x) = \begin{cases} 1, & x > (\hat{\mu}_0 + \hat{\mu}_1)/2 \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

A.2 Expected target generalization error of pooled FLD

We derive an expression for the expected target generalization error of the FLD decision rule when it is trained on the pooled data $\mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_o$ comprising of target and OOD data. For simplicity, we set the variance σ^2 of the class conditional densities of the synthetic data to 1.

The estimate μ_k is given by,

$$\begin{aligned} \hat{\mu}_k = \frac{1}{|\mathcal{D}_k|}\sum_{(x,y)\in\mathcal{D}_k} x &= \frac{\sum_{(x,y)\in\mathcal{D}_{t,k}} x + \sum_{(x,y)\in\mathcal{D}_{o,k}} x}{n_k + m_k} \\ &= \frac{n_k\bar{x}_{t,k} + m_k\bar{x}_{o,k}}{n_k + m_k} \\ &= \frac{n\bar{x}_{t,k} + m\bar{x}_{o,k}}{n + m}. \end{aligned} \tag{A.4}$$

where $\mathcal{D}_k$ is the set of samples of $\mathcal{D}$ that belongs to class k, $n_k = |\mathcal{D}_{t,k}|$ and $m_k = |\mathcal{D}_{o,k}|$ for $k \in \{0, 1\}$. $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ denote the sample means of class k in target and OOD datasets respectively. We assume that $\pi = \frac{1}{2}$ from which it follows that $n_k = n\pi_k = \frac{n}{2}$ and $m_k = m\pi_k = \frac{m}{2}$. We cannot explicitly compute $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ when the OOD samples are not explicitly known, because we cannot separate target samples from OOD samples in $\mathcal{D}$.

Since the samples are drawn from Gaussians, their averages also follow Gaussian distributions. Hence, the threshold $\hat{c} = \frac{\hat{\mu}_0 + \hat{\mu}_1}{2}$ of the hypothesis $\hat{h}$, estimated using FLD, is a random variable with a Gaussian distribution i.e., $\hat{c} \sim \mathcal{N}(\mu_h, \sigma_h^2)$ where

$$\mu_h = \mathbb{E}[\hat{c}] = \frac{m\Delta}{n+m},$$
$$\sigma_h^2 = \mathrm{Var}[\hat{c}] = \frac{1}{n+m}.$$

Let $f_{t,k}$ be the class conditional density of class k in the target distribution. Then, the target error of a hypothesis $\hat{h}$ is

$$\begin{aligned}
\mathbb{P}\left[\hat{h}(X) \neq y \mid \hat{c}\right] &= \frac{1}{2}\mathbb{P}_{x\sim f_{t,1}}[x < \hat{c}] + \frac{1}{2}\mathbb{P}_{x\sim f_{t,0}}[x > \hat{c}] \\
&= \frac{1}{2} + \frac{1}{2}\mathbb{P}_{x\sim f_{t,1}}[x < \hat{c}] - \frac{1}{2}\mathbb{P}_{x\sim f_{t,0}}[x < \hat{c}] \\
&= \frac{1}{2}\Big[1 + \Phi(\hat{c} - \mu) - \Phi(\hat{c} + \mu)\Big] \qquad \text{(A.5)}
\end{aligned}$$

Using Eq. (A.5), the expected target error $e_t(\hat{h}) = \mathbb{E}_{\hat{c}\sim\mathcal{N}(\mu_h,\sigma_h^2)}\left[\mathbb{P}\left[\hat{h}(x) \neq y \mid x, \hat{c}\right]\right]$ is

given by,

$$\begin{aligned} e_t(\hat{h}) &= \int_{-\infty}^{\infty} \frac{1}{2}\Big[1 + \Phi(\hat{c} - \mu) - \Phi(\hat{c} + \mu)\Big] \frac{1}{\sigma_h} \phi\left(\frac{\hat{c} - \mu_h}{\sigma_h}\right) d\hat{c} \\ &= \int_{-\infty}^{\infty} \frac{1}{2}\Big[1 + \Phi(y\sigma_h + \mu_h - \mu) - \Phi(y\sigma_h + \mu_h + \mu)\Big] \phi(y) dy \\ &= \frac{1}{2}\left[\Phi\left(\frac{\mu_h - \mu}{\sqrt{1+\sigma_h^2}}\right) + \Phi\left(\frac{-\mu_h - \mu}{\sqrt{1+\sigma_h^2}}\right)\right] \end{aligned}$$

In the last equality, we make use of the identity $\int_{-\infty}^{\infty} \Phi(cx + d)\phi(x)dx = \Phi\Big(\frac{d}{\sqrt{1+c^2}}\Big)$ where ϕ and Φ are the PDF and CDF of the standard normal. Substituting the expressions for μ_h, σ_h^2 into the above equation, we get

$$e_t(\hat{h}) = \frac{1}{2}\left[\Phi\left(\frac{m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right) + \Phi\left(\frac{-m\Delta - (n+m)\mu}{\sqrt{(n+m)(n+m+1)}}\right)\right] \tag{A.6}$$

For synthetic data with $\sigma^2 \neq 1$, the target generalization error can be obtained by simply replacing μ and Δ with $\frac{\mu}{\sigma}$ and $\frac{\Delta}{\sigma}$ respectively in Eq. (A.6).

A.3 Weighted Fisher's Linear Discriminant

We consider a target dataset $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^n$ and an OOD dataset $\mathcal{D}_o = \{(x_i, y_i)\}_{i=1}^m$. This setting differs from Section A.2 since we know whether each sample from $\mathcal{D} = \mathcal{D}_t \cup \mathcal{D}_o$ is OOD or not. This difference allows us to consider a log-likelihood function that weights the target and OOD samples differently, i.e. we consider

$$\ell(\mu_0, \mu_1, \sigma_0^2, \sigma_1^2) = \sum_{k=0}^{1} \left(\alpha \sum_{(x,y)\in D_{t,k}} \left[-\log \sigma_k - \frac{(x-\mu_k)^2}{2\sigma_k^2} \right] + (1-\alpha) \sum_{(x,y)\in D_{o,k}} \left[-\log \sigma_k - \frac{(x-\mu_k)^2}{2\sigma_k^2} \right] \right) + \text{constant} . \tag{A.7}$$

α is a weight that controls the contribution of the OOD samples in the log-likelihood function. Under the above log-likelihood, the maximum likelihood estimate for μ_k is

$$\hat{\mu}_k = \frac{\alpha \sum_{(x,y)\in D_{t,k}} x + (1-\alpha) \sum_{(x,y)\in D_{o,k}} x}{\alpha |D_{t,k}| + (1-\alpha)|D_{o,k}|}. \tag{A.8}$$

We can make use of the above $\hat{\mu}_k$ to get a weighted FLD decision rule using Eq. (A.3).

A.4 Expected target generalization error of weighted FLD

Let $\sigma^2 = 1$. We re-write $\hat{\mu}_k$ from Eq. (A.8) using notation from Section A.2:

$$\hat{\mu}_k = \frac{n\alpha \bar{x}_{t,k} + m(1-\alpha)\bar{x}_{o,k}}{n\alpha + m(1-\alpha)}.$$

We can explicitly compute $\bar{x}_{t,k}$ and $\bar{x}_{o,k}$ in the OOD-aware setting since we can separate target samples from OOD samples. For the synthetic distribution, the threshold $\hat{c}_\alpha = \frac{\hat{\mu}_0 + \hat{\mu}_1}{2}$ of the hypothesis $\hat{h}_\alpha$ follows a normal distribution $\mathcal{N}(\mu_{h\alpha}, \sigma_{h\alpha}^2)$ where

$$\mu_{h\alpha} = \mathbb{E}[\hat{c}_\alpha] = \frac{m(1-\alpha)\Delta}{n\alpha + m(1-\alpha)}$$

$$\sigma_{h\alpha}^2 = \mathrm{Var}[\hat{c}_\alpha] = \frac{\alpha^2 n + (1-\alpha)^2 m}{(\alpha n + (1-\alpha)m)^2}$$

Similar to the Section A.2, we derive an analytical expression for the expected target risk of the weighted FLD, which is

$$e_t(\hat{h}_\alpha) = \frac{1}{2}\left[\Phi\left(\frac{\mu_{h\alpha} - \mu}{\sqrt{1+\sigma_{h\alpha}^2}}\right) + \Phi\left(\frac{-\mu_{h\alpha} - \mu}{\sqrt{1+\sigma_{h\alpha}^2}}\right)\right] \tag{A.9}$$

A.5 Experiments with Neural Networks

A.5.1 Datasets

We experiment on images from CIFAR-10, CINIC-10 (Darlow et al., 2018) and several datasets from the DomainBed benchmark (Gulrajani and Lopez-Paz, 2020): Rotated MNIST (Ghifary et al., 2015), PACS (Li, Y. Yang, Y.-Z. Song, and Timothy M Hospedales, 2017a), and DomainNet (Peng et al., 2019). We construct sub-tasks from these datasets as explained below.

CIFAR-10 We use of tasks from Split-CIFAR10 (Zenke et al., 2017) which are five binary classification sub-tasks constructed by grouping consecutive labels of CIFAR-10. The 5 task distributions are airplane vs. automobile (T_1), bird vs. cat (T_2), deer vs. dog (T_3), frog vs. horse (T_4) and ship vs truck (T_5). All the images are of size $(3, 32, 32)$.

CINIC-10 This dataset combines CIFAR-10 with downsampled images from ImageNet. It contains images of size $(3, 32, 32)$ across 10 classes (same classes as CIFAR-10). As there are two sources of the images within this dataset, it is a natural candidate for studying distribution shift. The construction of the dataset motivates us to consider two distributions from CINIC-10: (1) Distribution with only CIFAR images, and (2) Distribution with only ImageNet images.

Rotated MNIST This dataset is constructed from MNIST by rotating the images (which are of size $(1, 28, 28)$. All MNIST images rotated by an angle θ° are considered to belong to the same distribution. Hence, we can consider the family of distributions which is characterized by 10-way classification of hand-written digit images rotated θ°. By varying θ, we can obtain a number of different distributions.

PACS PACS contains images of size $(3, 224, 244)$ with 7 classes present across 4 domains {art, cartoons, photos, sketches}. In our experiments, we consider only 3 classes ({Dog, Elephant, Horse}) out of the 7 and consider the 3-way classification of images from a given domain as a distribution. Therefore, we can have a total of 4 distinct distributions from PACS.

DomainNet Similar to PACS, this dataset contains images of size $(3, 224, 244)$ from 6 domains {clipart, infograph, painting, quickdraw, real, sketches} across 345 classes. In our experiments, we consider only 2 classes, ({Bird, Plane}) and consider the binary classification of images from a given domain as a distribution. As a result, we can have a total of 6 distinct distributions from PACS.

A.5.2 Forming Target and OOD Distributions

We consider two types of setups to study the impact of OOD data:

OOD data arising due to geometric intra-class nuisances We study the effect of intra-class nuisances using a classification task using samples from a target distribution and OOD samples from a transformed version of the same distribution. In this regard, we consider the following experimental setups.

1. **Rotated MNIST: unrotated images as target and θ°- rotated images as OOD:** We consider the 10-way classification (see Section A.5.1) of unrotated images as the target data and that of the θ rotated images as the OOD data. We can have different OOD data by selecting different values for θ.

2. **Rotated CIFAR-10: $\boldsymbol{T_2}$ as target and rotated $\boldsymbol{T_2}$ as OOD:** We choose the bird vs. cat (T_2) task from Split-CIFAR10 as the target distribution. We then rotate the images of T_2 by an angle θ° counter-clockwise around their centers to form a new task distribution denoted by θ-T_2, which we consider as OOD. Different OOD datasets can be obtained by selecting different values for θ.

3. **Blurred CIFAR-10: $\boldsymbol{T_4}$ as target and blurred $\boldsymbol{T_4}$ as OOD:** We choose the Frog vs. Horse (T_4) task from Split-CIFAR10 as the target distribution. We then add Gaussian blur with standard deviation σ to the images of T_4 to form a new task distribution denoted by σ-T_2, which we consider as the OOD. By setting distinct values for σ, we have different OOD datasets.

OOD data arising due to category shifts and concept drifts We study this aspect using two different target and OOD classification problems as described below.

1. **Split-CIFAR10: T_i as Target and T_j as OOD:** We choose a pair of distinct tasks from the 5 binary classification tasks of Split-CIFAR10 and consider one as the target distribution and the other as the OOD. We perform experiments for all pairs of distributions (20 in total) in Split-CIFAR10.

2. **PACS: Photo-domain as target and X-domain as OOD:** Out of the four 3-way classification tasks from PACS described in Section A.5.1, we select the photo-domain as the target distribution and consider one of the remaining 3 domains (for instance, the sketch-domain) as the OOD.

3. **DomainNet: Real-domain as target and X-domain as OOD:** Out of the six binary classification tasks from DomainNet described in Section A.5.1, we consider the real-domain as the target distribution and select one of the remaining 5 domains (for instance, the painting-domain) as OOD.

4. **CINIC-10: CIFAR10 as target and ImageNet as OOD:** Here we simply select the 10-way classification of CIFAR images as the target distribution and that of ImageNet as OOD.

A.5.3 Experimental Details

In the above experiments, for each random seed, we randomly select a fixed sample of size n from the target distribution. Next, we select OOD samples of varying sizes m

such that the previous samples are a subset of the next set of samples. The samples from both target and OOD distributions preserve the ratio of the classes. For rotated MNIST, rotated CIFAR-10, and blurred CIFAR-10, when selecting multiple sets of OOD samples, the OOD images that correspond to the n selected target images are disregarded. For PACS and DomainNet, the images are downsampled to $(3, 64, 64)$ during training.

For both the OOD-agnostic (OOD unknown) and OOD-aware (OOD known) settings, at each m-value, we construct a combined dataset containing the n sized target set and m sized OOD set. We use a CNN (see Section A.5.4) for experiments in the both of these settings. We experiment with α fixed to 0.5 (naive OOD-aware model) and with the optimal α^*. We average the runs over 10 random seeds and evaluate on a test set comprised of only target samples.

In the optimal OOD-aware setting, we use a grid-search to find the optimal α^* for each value of m. We use an adaptive equally-spaced α search set of size 10 such that it ranges from α^*_{prev} to 1.0 (excluding 1.0) where α^*_{prev} is the optimal value of α corresponding to the previous value of m. We use this search space since we expect α^* to be an increasing function of m.

A.5.4 Neural Architectures and Training

We primarily use 3 different network architectures in our experiments: (a) a small convolutional network with 0.12M parameters (denoted by SmallConv), (b) a wide residual network (Zagoruyko and Komodakis, 2016) of depth 10 and widening factor 2 (WRN-10-2), and (c) a larger wide residual network of depth 16 and widening factor

4 (WRN-16-4). SmallConv comprises of 3 convolution layers (kernel size 3 and 80 filters) interleaved with max-pooling, ReLU, batch-norm layers, with a fully-connected classifier layer in our experiments.

Table A.1 provides a summary of network architectures used in the experiments described earlier. All the networks are trained using stochastic gradient descent (SGD) with Nesterov's momentum and cosine-annealed learning rate. The hyperparameters used for the training are, learning rate of 0.01, and a weight-decay of 10^{-5}. All the images are normalized to have mean 0.5 and standard deviation 0.25. In the OOD-agnostic setting, we use sampling without replacement to construct the mini-batches. In the OOD-aware settings (both naive and optimal), we construct mini-batches with a fixed ratio of target and OOD samples. See Section A.5.5 and Fig. A.1 for more details.

Experiment	**Network(s)**	**# classes**	**n**	**Image Size**	**Mini-Batch Size**
Rotated MNIST	SmallConv	10	100	(1,28,28)	128
Rotated CIFAR-10	SmallConv, WRN-10-2	2	100	(3,32,32)	128
Blurred CIFAR-10	WRN-10-2	2	100	(3,32,32)	128
Split-CIFAR10	SmallConv, WRN-10-2	2	100	(3,32,32)	128
PACS	WRN-16-4	3	30	(3,64,64)	16
DomainNet	WRN-16-4	2	50	(3,64,64)	16
CINIC-10	WRN-10-2	10	100	(3,32,32)	128

Table A.1: Summary of network architectures used in the experiments

A.5.5 Construction of Mini-Batches

Consider a mini-batch $\{(x_{b_i}, y_{b_i})\}_{i=1}^{B}$ of size B. Let the randomly chosen mini-batch contains B_t target samples and B_o OOD samples ($B = B_t + B_o$). Let $\hat{e}_{B,t}(h)$ and

$\hat{e}_{B,o}(h)$ denote the average mini-batch surrogate losses for the B_t target samples and B_o OOD samples respectively.

In the OOD-aware (when we know which samples are OOD) setting, $\hat{e}_{B,t}(h)$ and $\hat{e}_{B,o}(h)$ can be computed explicitly for each mini-batch resulting in the mini-batch gradient

$$\hat{\nabla}\hat{e}_B(h) = \alpha\hat{\nabla}\hat{e}_{B,t}(h) + (1-\alpha)\hat{\nabla}\hat{e}_{B,o}(h). \tag{A.10}$$

If we were to sample without replacement, we expect the fraction of the target samples in every mini-batch to approximately equal $\frac{n}{n+m}$ on average. However, if $m >> n$, we run into a couple of issues. First, we observe that most mini-batches have no target samples, making it impossible to compute $\hat{\nabla}\hat{e}_{B,t}(h)$. Next, even if the mini-batch does have some target samples, there are very few of them, resulting in high variance in the estimate $\hat{\nabla}\hat{e}_{B,t}(h)$.

Hence, we find it beneficial to consider alternative sampling schemes for the mini-batch. Independent of the values of n and m, we use a sampler which ensures that every mini-batch has a fixed fraction of target samples, which we denote by β. For example if the mini-batch size B is 20 and if $\beta = 0.5$, then every mini-batch has 10 target samples and 10 OOD samples regardless of n and m. Note that this sampling biases the gradient, but results in reduced variance estimates. In practice, we observe improved test errors when we set β to either 0.5 or 0.75.

A.5.6 Additional Experiments with Neural Networks

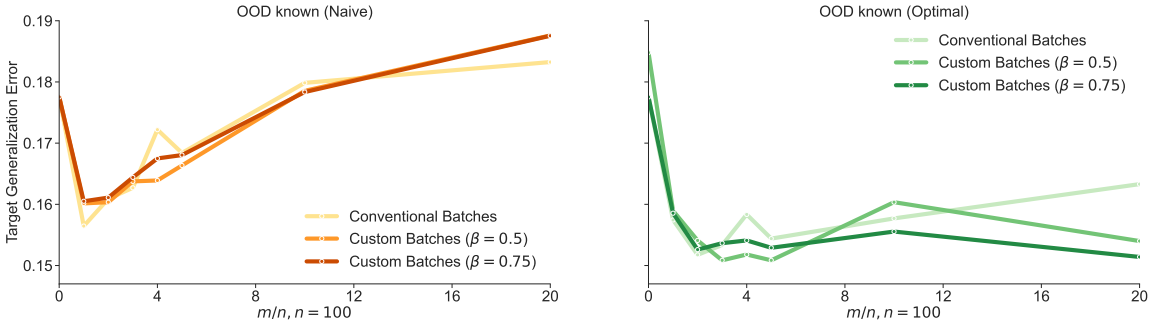


Figure A.1: Standard mini-batching strategy versus ensuring that every mini-batch has a fraction β samples from the target distribution. The test error of a neural network (SmallConv) on the target distribution (Y-axis) is plotted against the number of OOD samples (X-axis) for the target-OOD pair of T_1 and T_5. One set of curves (lightest shade of green and yellow) considers mini-batches which are constructed using sampling without replacement; This is the standard strategy used in supervised learning. The other curves consider $\beta = 0.5$ (intermediate shades of orange and green) and $\beta = 0.75$ (darkest shade of red and green). All plots are in the OOD-aware setting. **Left:** If we consider $\alpha = 0.5$, then the choice of β has little effect on the generalization error. **Right:** However, if we use α^* to weight the OOD and target losses, then the generalization error depends on the the choice of β with $\beta = 0.75$ having the lowest test error.

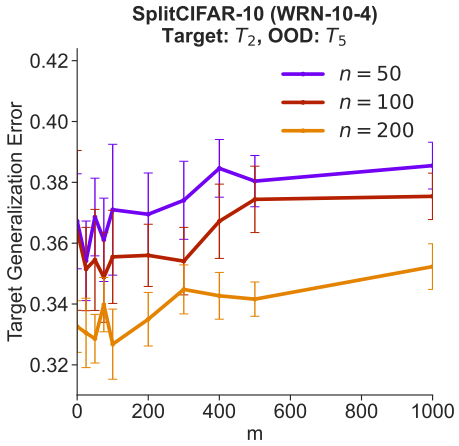


Figure A.2: We plot the generalization error on the target distribution (Y-axis) against the number of OOD samples m (X-axis) across three different target sample sizes, $n = 50, 100$ and 200 for the target-OOD pair T_2 and T_5 from Split-CIFAR10. Non-monotonic trends in generalization error are present in all the three cases. The trend is less apparent for $n = 50$ since the number of samples is small resulting in a large variance. Error bars indicate 95% confidence intervals (10 runs).

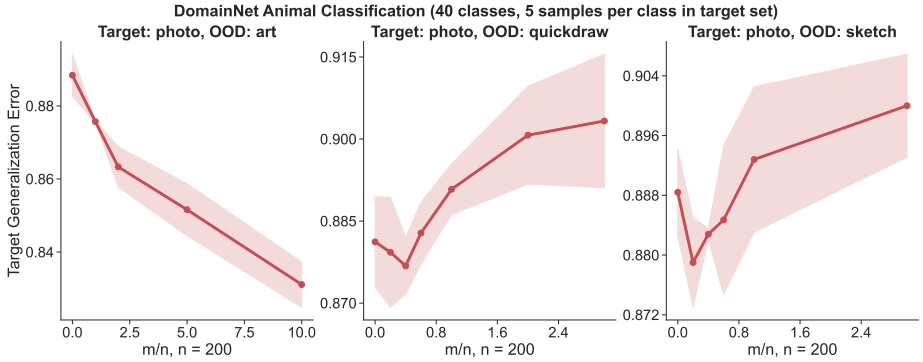
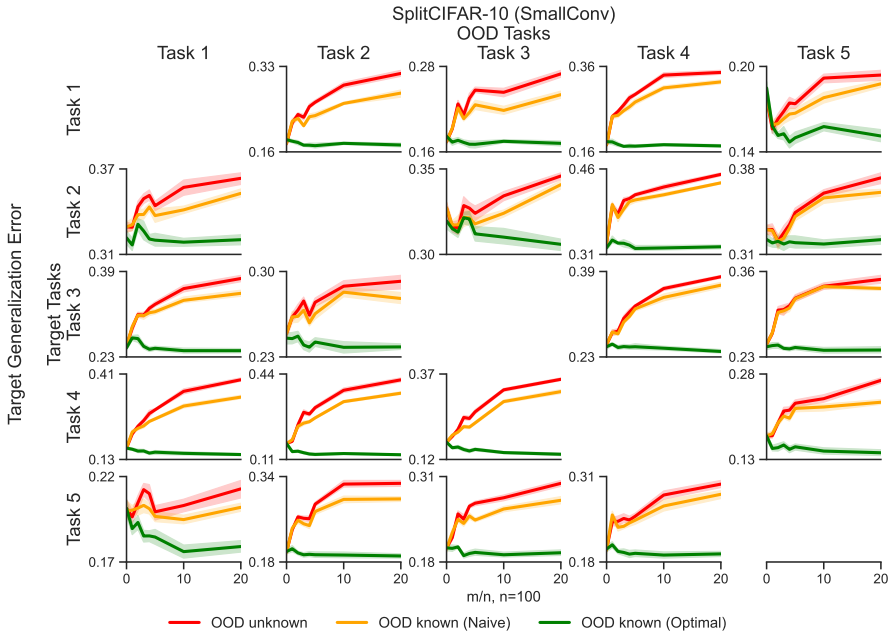
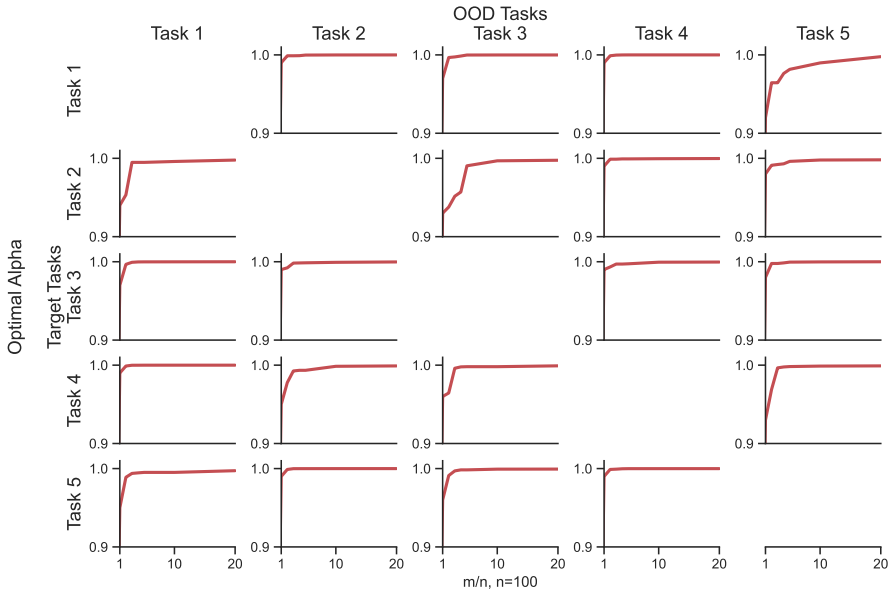


Figure A.3: We consider a 40-class classification problem from DomainNet where the classes are animals from three super-classes: mammals, cold blooded animals and birds. The target distribution considers images of animals from the "real" domain. OOD data considers images from the domains "paintings", "quickdraw" and "sketches". We plot the target generalization error against the ratio of OOD and target samples and observe the risk to be non-monotonic for 2 of the 3 OOD domains. Note that the error of the trained network (0.85) is lower than the error of a classifier that predicts all classes with uniform probability (0.975). The error is high because we use very few training samples; the number of target samples is 200 (i.e. only 5 samples per class). Note that the error bars indicate 95% confidence intervals over 3 runs.

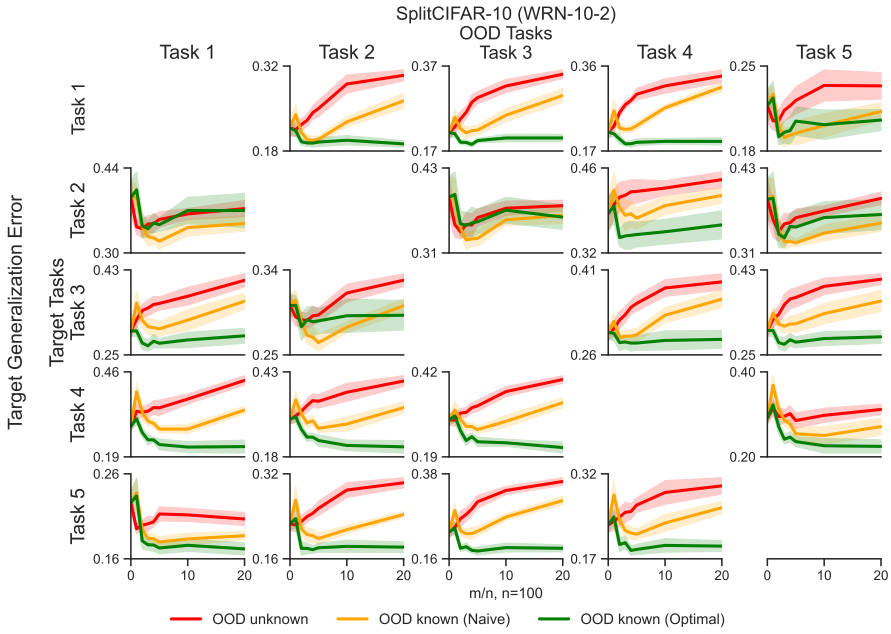


(a)

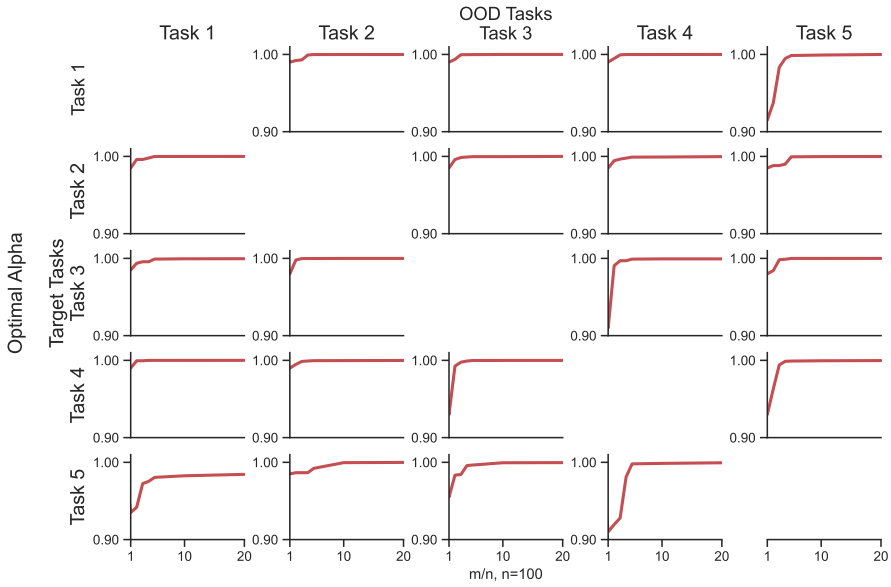


(b)

Figure A.4: For all target-OOD pairs from Split-CIFAR10, we plot **(a)** the test error of SmallConv on the target distribution (Y-axis) against the ratio of number of OOD samples to the number of samples from the target task (X-axis), **(b)** the optimal α^* (Y-axis) against the number of OOD samples (X-axis) for the optimally weighted OOD-aware setting.



(a)



(b)

Figure A.5: For all target-OOD pairs from Split-CIFAR10, we plot **(a)** the test error of WRN-10-2 on the target distribution (Y-axis) against the ratio of number of OOD samples to the number of samples from the target task (X-axis), **(b)** the optimal α^* (Y-axis) against the number of OOD samples (X-axis) for the optimally weighted OOD-aware setting.

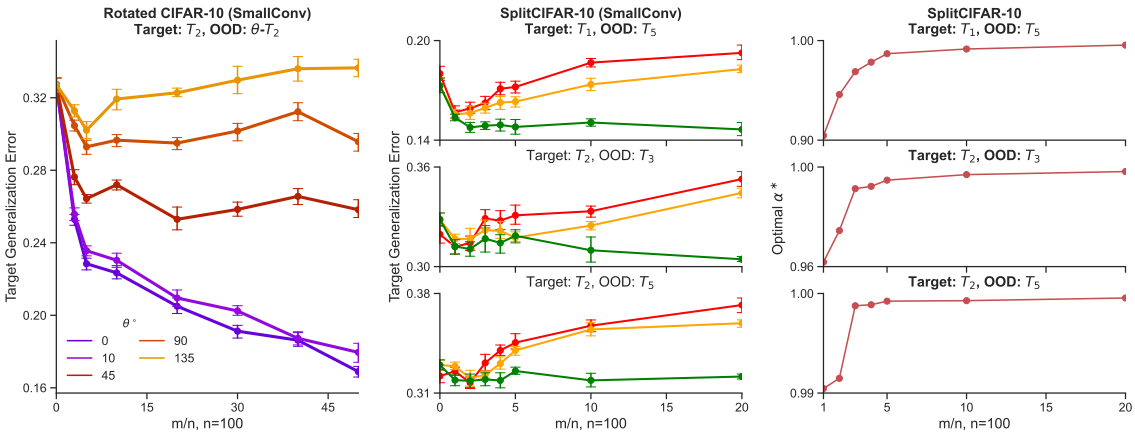


Figure A.6: Left: A binary classification problem (Bird vs. Cat) is the target distribution and images of these classes rotated by different angles θ° are OOD. We see non-monotonic curves for larger values of θ°. For 135° in particular, the generalization error at $m/n = 50$ is worse than the generalization error with no OOD samples, i.e. OOD samples actively hurt generalization.

Middle: Generalization error on the target distribution is plotted against the number of OOD samples for 3 different target-OOD pairs constructed from CIFAR-10 for three settings: OOD-agnostic ERM where we minimize the total average risk over both distributions (red), an objective which minimizes the sum of the average loss of the target and OOD distributions which corresponds to $\alpha = 1/2$ (OOD-aware, yellow) and an objective which minimizes an optimally weighted convex combination of the target and OOD empirical loss (green).

Right: The optimal α^* obtained via grid search for the three problems in the middle column plotted against different number of OOD samples. Note that the appropriate value of α lies very close to 1 but it is never exactly 1. In other words the OOD samples always benefit if we use the weighted objective, even if this benefit is marginal in cases when OOD samples are very different from those of the target.

A.6 Derivation of the Ben-David, Blitzer, et al. (2010) upper bound for the OOD-agnostic pooled FLD

We begin by defining the following quantities: Given a hypothesis $h : \mathcal{X} \to \{0, 1\}$, the probability according to the distribution P_t that h disagrees with a labeling function f is defined as,

$$e_t(h, f) = \mathbb{E}_{x \sim P_t} \left[|h(x) - f(x)| \right]$$

For a hypothesis space $\mathcal{H}$, (Ben-David, Blitzer, et al., 2010) defines the divergence measure between two distributions P_t and P_o in the symmetric difference hypothesis space as,

$$d^*_{\mathcal{H}}(P_t, P_o) = 2 \sup_{h, h' \in \mathcal{H}} |e_o(h, h') - e_t(h, h')|$$

With these definitions in place, we restate a slightly modified version of the Theorem 3 from (Ben-David, Blitzer, et al., 2010) below.

Theorem A.6.1. Let $\mathcal{H}$ be a hypothesis space of VC dimension d. Let $\mathcal{D}$ be a dataset generated by drawing n samples from a target distribution P_t and m OOD samples from P_o. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\alpha e_t(h) + (1 - \alpha) e_o(h)$ on D and $h^*_t = \min_{h \in \mathcal{H}} e_t(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$, with probability as least $1 - \delta$ (over the choice of samples),

$$\begin{aligned} e_t(\hat{h}) \leq e_t(h^*_t) + 4\sqrt{\frac{\alpha^2}{n} + \frac{(1-\alpha)^2}{m}} \sqrt{2d \cdot \log(2(n+m+1)) + 2\log\left(\tfrac{8}{\delta}\right)} \\ + 2(1-\alpha)\left(\frac{1}{2} d_{\mathcal{H}}(P_t, P_o) + \lambda\right) \end{aligned} \tag{A.11}$$

where λ is the combined error of the ideal joint hypothesis given by $h^* = \text{argmin}_{h \in H}\, e_t(h) + e_s(h)$. Hence, $\lambda = e_t(h^*) + e_s(h^*)$.

We wish to adapt the above theorem to the pooled FLD and consequently find an expression for the upper bound in terms of n, m and Δ. As we do not know of the existence of OOD samples in dataset $\mathcal{D}$, we find the hypothesis $\hat{h}$ by minimizing the empirical loss below.

$$\begin{aligned}\hat{e}(h) &= \frac{1}{n+m}\sum_{i=1}^{n+m}\ell(h(x_i), y_i) \\ &= \frac{1}{n+m}\sum_{(x,y)\in\mathcal{D}_t}\ell(h(x), y) + \frac{1}{n+m}\sum_{(x,y)\in\mathcal{D}_o}\ell(h(x), y) \\ &= \frac{n}{n+m}e_t(h) + \frac{m}{n+m}e_o(h).\end{aligned}$$

Here, we have assumed that $\ell(\cdot)$ is the 0-1 loss. Therefore, under the OOD agnostic setting, we minimize the objective function $e(h) = \alpha e_t(h) + (1-\alpha)e_o(h)$ where $\alpha = n/(n+m)$. Since we deal with a univariate FLD, the VC dimension of the hypothesis space is equal to $d = 1 + 1 = 2$. Plugging these terms in Eq. (A.11), we can rewrite the upper bound as,

$$e_t(\hat{h}) \leq e_t(h_t^*) + 4\sqrt{4\log(2(n+m+1)) + 2\log(\frac{8}{\delta})} + \frac{2m}{n+m}\left(\frac{1}{2}d_{\mathcal{H}}(P_t, P_o) + \lambda\right) \quad \text{(A.12)}$$

The first term of the above expression corresponds to the error of the best hypothesis h_t^* in class $\mathcal{H}$ for the target distribution P_t. Thus, $e_t(h_t^*)$ is equivalent to the Bayes optimal error or the lowest possible error achievable for the target distribution, under $\mathcal{H}$. By setting $m = 0$ in Eq. (A.6), we arrive at the expected error $e_t(\hat{h})$ on the target

distribution when we estimate $\hat{h}$ using n target samples. The Bayes optimal error $e_t(h_t^*)$ is then equal to the limit of $e_t(\hat{h})$ as $n \to \infty$.

$$e_t(h_t^*) = \lim_{n\to\infty} e_t(\hat{h}) = \lim_{n\to\infty} \Phi\left(- \frac{n(\mu/\sigma)}{\sqrt{n(n+1)}}\right) = \Phi(-\mu/\sigma)$$

Intuitively, the threshold corresponding to the ideal joint hypothesis h^* for the FLD is given by the mid point between the centers of the two distributions,

$$h^*(x) = \operatorname*{argmin}_{h\in\mathcal{H}} e_o(h) + e_t(h) = \mathbb{1}_{(\Delta/2,\infty)}(x)$$

where $I_A(x)$ is the indicator function of the subset A. Therefore, the combined error λ of the ideal joint hypothesis can be computed as follows.

$$\begin{aligned}
\lambda &= e_o(h^*) + e_t(h^*) \\
&= \frac{1}{2}\mathbb{P}_{x\sim f_{t,0}}[x > \Delta/2] + \frac{1}{2}\mathbb{P}_{x\sim f_{t,1}}[x < \Delta/2] + \frac{1}{2}\mathbb{P}_{x\sim f_{o,0}}[x > \Delta/2] + \frac{1}{2}\mathbb{P}_{x\sim f_{o,1}}[x < \Delta/2] \\
&= \Phi\left(\frac{-\Delta/2-\mu}{\sigma}\right) + \Phi\left(\frac{\Delta/2-\mu}{\sigma}\right)
\end{aligned}$$

Finally, we turn to the divergence term $d_{\mathcal{H}}(P_t, P_o)$. Let $h, h' \in \mathcal{H}$ be two hypotheses with thresholds c and c', respectively. From the definition of $e_t(h, h')$ we have,

$$\begin{aligned}
e_t(h, h') &= \mathbb{E}_t\left[|h(x) - h'(x)|\right] \\
&= \mathbb{E}_t\left[\left|\mathbb{1}_{(c,\infty)}(x) - \mathbb{1}_{(c',\infty)}(x)\right|\right] \\
&= \mathbb{E}_t\left[\mathbb{1}_{(\min(c,c'),\max(c,c')]}(x)\right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{P}_t[\min(c, c') < x \leq \max(c, c')] \\
&= \frac{1}{2}\mathbb{P}[x \leq \max(c, c') \mid y = 0] + \frac{1}{2}\mathbb{P}[x \leq \max(c, c') \mid y = 1] \\
&\quad - \frac{1}{2}\mathbb{P}[x \leq \min(c, c') \mid y = 0] - \frac{1}{2}\mathbb{P}[x \leq \min(c, c') \mid y = 1] \\
&= \frac{1}{2}\Bigg[\Phi\left(\frac{\max(c, c') + \mu}{\sigma}\right) + \Phi\left(\frac{\max(c, c') - \mu}{\sigma}\right) \\
&\qquad - \Phi\left(\frac{\min(c, c') + \mu}{\sigma}\right) - \Phi\left(\frac{\min(c, c') - \mu}{\sigma}\right)\Bigg] \\
&= \psi_{\mu,\sigma}(c, c')
\end{aligned}
$$

Similarly, we can show that $e_o(h, h') = \psi_{\mu,\sigma}(c - \Delta, c' - \Delta)$. Therefore, we can rewrite the expression for $d_{\mathcal{H}}(P_t, P_o)$ as follows.

$$
d_{\mathcal{H}}(P_t, P_o) = 2 \sup_{h,h' \in \mathcal{H}} |e_o(h, h') - e_t(h, h')| = 2 \sup_{c,c' \in [0,\Delta]} |\psi_{\mu,\sigma}(c - \Delta, c' - \Delta) - \psi_{\mu,\sigma}(c, c')| = d^*_{\mathcal{H}}(\Delta)
$$

Using this expression we can numerically compute d^*_H, given the values of μ, σ and Δ. Plugging in the expressions we have obtained for $e_t(h^*_t)$, λ and $d_{\mathcal{H}}(P_t, P_o)$ in Eq. (A.12), we arrive at the desired upper bound for the expected target error $e_t(\hat{h})$ of our FLD example.

$$
\begin{aligned}
e_t(\hat{h}) \leq\ & \Phi(-\mu/\sigma) + 4\sqrt{4\log(2(n + m + 1)) + 2\log\left(\tfrac{8}{\delta}\right)} \\
& + \frac{2m}{n + m}\left[\frac{1}{2}d^*_{\mathcal{H}}(\Delta) + \Phi\left(\frac{-\Delta/2 - \mu}{\sigma}\right) + \Phi\left(\frac{\Delta/2 - \mu}{\sigma}\right)\right]
\end{aligned}
\tag{A.13}
$$

In Figs. A.7 and A.8, we compare the upper bound and the true target generalization error. The bound is computed via Eq. (A.13) and the true error is computed analytically

using Eq. (A.6). Across all settings, the bound is significantly vacuous and fails to capture the non-monotonic trends observed in the true error. This suggests that the theoretical framework of Ben-David, Blitzer, et al. (2010) is insufficient to explain the phenomenon identified in this work.

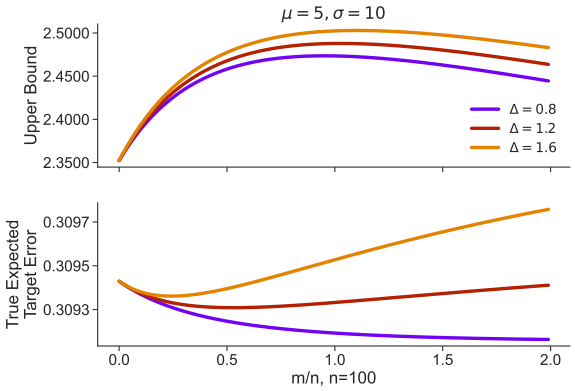


Figure A.7: True target error (**bottom**) and generalization error upper bound (**top**) vs. m/n for the FLD Gaussian example ($\mu = 5, \sigma = 10$). The bound is vacuous and fails to capture the non-monotonic trend, though for large distribution shift Δ its shape becomes consistent with the true error.

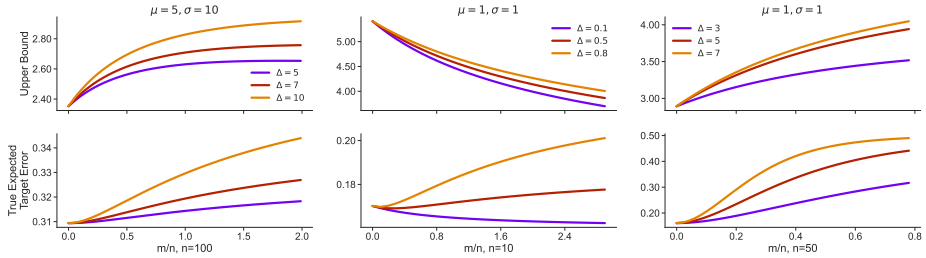


Figure A.8: Upper bound and true target error vs. m/n for three FLD variants. The bound's shape roughly agrees with the true error when distribution shift Δ is large (left and right columns), but fails to capture non-monotonic trends (middle column and Fig. A.7). The bound is vacuous in all cases, suggesting that Ben-David, Blitzer, et al. (2010) upper bound does not explain the non-monotonic behavior identified in this work.

A.7 Physiological prediction tasks

A.7.1 EEG-based cognitive load classification

The first physiological prediction dataset was collected under NASA's Multi-Attribute Task Battery II (MATB-II) protocol, which induces four levels of cognitive load, passive, low, medium, and high, by varying the number of tasks a participant must actively tend to. The study included 50 healthy participants (29 female, 21 male, ages 18–39, mean 25.9, standard deviation 5.4 years). Each participant completed two sessions of approximately 50 minutes each, separated by a 10-minute break, for a total of 100 sessions. Each session contained three segments divided into blocks with the four cognitive load levels.

EEG data was recorded using a 24-channel Smarting MOBI device and preprocessed with a 0.5 Hz high-pass filter and a 30 Hz low-pass filter. The continuous signal was segmented into ten-second, non-overlapping windows. For each window, the spectral power in three frequency bands, theta (4–8 Hz), alpha (8–12 Hz), and lower beta (12–20 Hz), was computed and normalized on a per-channel basis. Only the frontal channels (Fp1, Fp2, F3, F4, F7, F8, Fz, aFz) were retained, yielding a $3 \times 8 = 24$-dimensional feature vector per window. The classification task was binary: low cognitive load (passive and low conditions) versus high cognitive load (medium and high conditions).

For each session, a proportion $p \in \{0.05, 0.1, 0.2, 0.5\}$ of the participant's windowed data was randomly sampled as target training data. The source projection vectors were computed from all other sessions except the target participant's other session, yielding 98 source projection vectors. The training data was used to estimate a trans-

lation and scaling to match the class-conditional means and covariance to the model assumptions of Section 2.3. The balanced accuracy and optimal convex coefficient α^* were computed over 100 different train–test splits per session, with α searched over the grid $\{0, 0.1, 0.2, \ldots, 1.0\}$. The empirical concentration parameter κ, estimated from all sessions' projection vectors, was approximately 17.2.

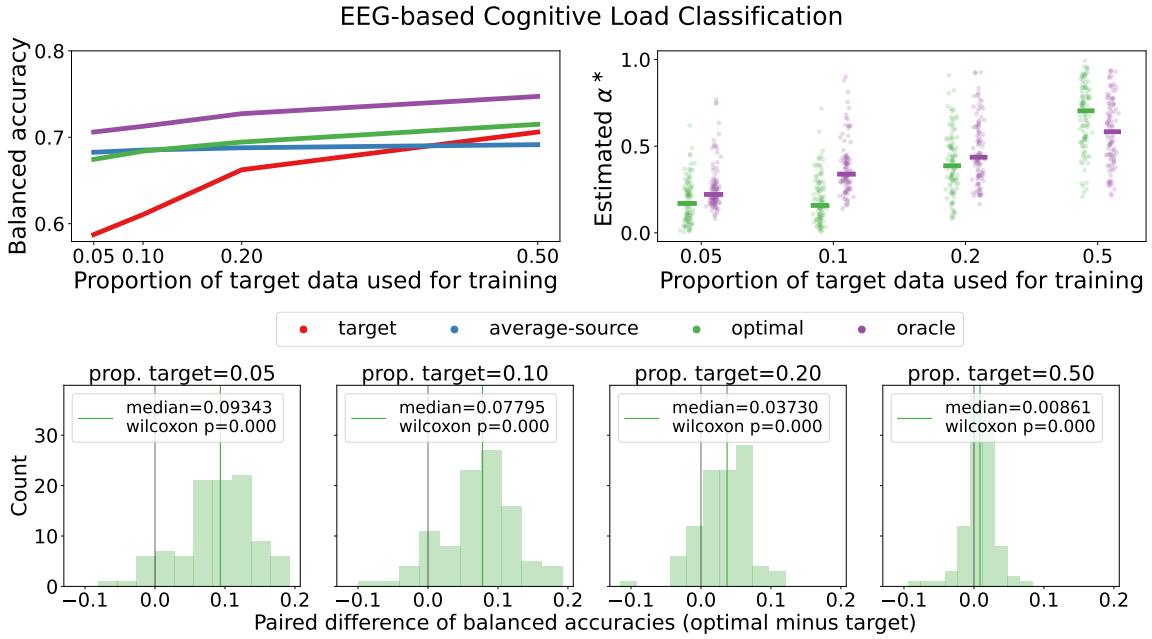


Figure A.9: Balanced accuracy and relevant convex coefficients (top) and relative performance of the optimal and target classifiers (bottom) for the MATB-II cognitive load classification task.

The approximately optimal classifier outperformed or matched both the target-only and average-source classifiers across the entire data regime. For $p = 0.05$, the optimal classifier achieved higher balanced accuracy than the target classifier in 92 of 100 sessions, with a median improvement of 9.3% and a maximum improvement of 19.2%. The advantage diminished as more target data became available, with the median improvement falling to 7.8% for $p = 0.1$, 3.7% for $p = 0.2$, and 0.9% for $p = 0.5$, but

the optimal classifier was never meaningfully worse than the target classifier at any proportion. The one-sided Wilcoxon signed-rank test rejected the null hypothesis of no improvement at $p < 0.001$ for all four data proportions. The high empirical $\kappa \approx 17.2$ in this dataset meant that the average-source classifier provided a strong low-variance proxy, and the optimal α^* accordingly favored the source at low p and shifted toward the target as p increased.

A.7.2 EEG-based stress classification

The second dataset comes from a mental arithmetic stress study. During the resting condition, participants counted mentally with eyes closed for three minutes. During the stress condition, participants were given a four-digit number and a two-digit number and asked to recursively subtract the latter from the former for four minutes, a procedure known to induce physiological stress.

The study initially enrolled 66 participants (47 female, 19 male) of matched age. Thirty participants were excluded due to poor EEG quality, leaving 36 participants in the released dataset. The EEG data was preprocessed with a high-pass filter and a 50 Hz power-line notch filter, and artifacts from eye movements and muscle tension were removed via independent component analysis (ICA). The preprocessed signal was segmented into 2.5-second non-overlapping windows. Only the centerline channels (Fz, Cz, Pz) were retained, and spectral power was computed in the theta, alpha, and lower beta bands, yielding a $3 \times 3 = 9$-dimensional feature vector per window. The classification task was binary: stressed versus not stressed.

The experimental protocol followed the same structure as the cognitive load task.

For each participant, a proportion $p \in \{0.05, 0.1, 0.2, 0.5\}$ of the data was used as target training data, and the projection vectors from the remaining 35 participants served as sources. The balanced accuracy and optimal α^* were computed over 100 train–test splits per participant. The empirical concentration parameter κ was less than 3, substantially lower than in the cognitive load task, reflecting greater variability in the projection vectors across participants.

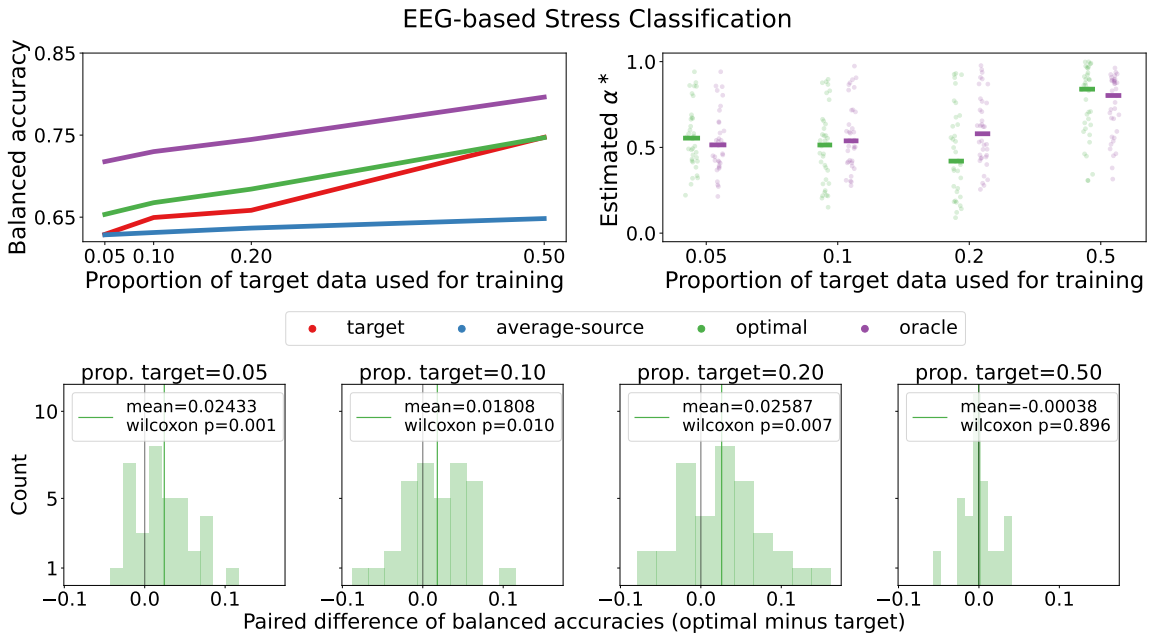


Figure A.10: Balanced accuracy and relevant convex coefficients (top) and relative performance of the optimal and target classifiers on a per-participant basis (bottom) for the Mental Math EEG-based stress classification task.

The optimal classifier outperformed the target classifier for the majority of participants in the low-data regime, though the margins were smaller than in the cognitive load task, which is consistent with the lower empirical κ. For $p = 0.05$, 0.1, and 0.2, the optimal classifier achieved higher balanced accuracy than the target classifier for 25, 24, and 24 of the 36 participants, respectively, with median improvements in the range

of 1.8–2.6% and maximum improvements of up to 19.2%. For $p = 0.5$, the distribution of paired differences was centered near zero, indicating that the target classifier had sufficient data to match the optimal classifier. The Wilcoxon test confirmed statistical significance for $p = 0.05$ (p-value $= 0.001$), $p = 0.1$ (p-value $= 0.01$), and $p = 0.2$ (p-value $= 0.007$), but not for $p = 0.5$ (p-value $= 0.896$). The poor performance of the average-source classifier throughout the regime reflects the low concentration of the source projection vectors.

A.7.3 ECG-based social stress classification

The third dataset is the WEarable Stress and Affect Detection (WESAD) dataset, which contains multimodal data collected while participants underwent a neutral baseline condition (passively reading a magazine for approximately 20 minutes) and a stress condition (a combination of the Trier Social Stress Test and a mental arithmetic task for approximately 10 minutes). Participants meditated between conditions. The analysis used 14 of the 15 participants and only their ECG data, recorded at 700 Hz.

The ECG signal was downsampled to 100 Hz and segmented into 15-second non-overlapping windows. Hamilton's peak detection algorithm was applied to each window to identify the inter-beat intervals. Three features were extracted per window: the proportion of successive inter-beat intervals differing by more than 20 ms, the normalized standard deviation of the interval lengths, and the ratio of high-frequency (15–40 Hz) to low-frequency (4–15 Hz) power in the interval waveform after applying a Lomb–Scargle correction for unevenly sampled data. These three heart-rate-variability features are known to carry discriminative information for stress detection, though

typically for longer time windows than the 15 seconds used here. The classification task was binary: baseline versus stress.

The experimental protocol was identical to the two EEG tasks. For each participant, a proportion $p \in \{0.05, 0.1, 0.2, 0.5\}$ of the data served as target training data, and the projection vectors from the remaining 13 participants served as sources. The balanced accuracy and optimal α^* were computed over 100 train–test splits. The empirical concentration parameter κ was approximately 1.5, the lowest of the three datasets, indicating high variability in the projection vectors and a correspondingly weaker contribution of the average-source classifier.

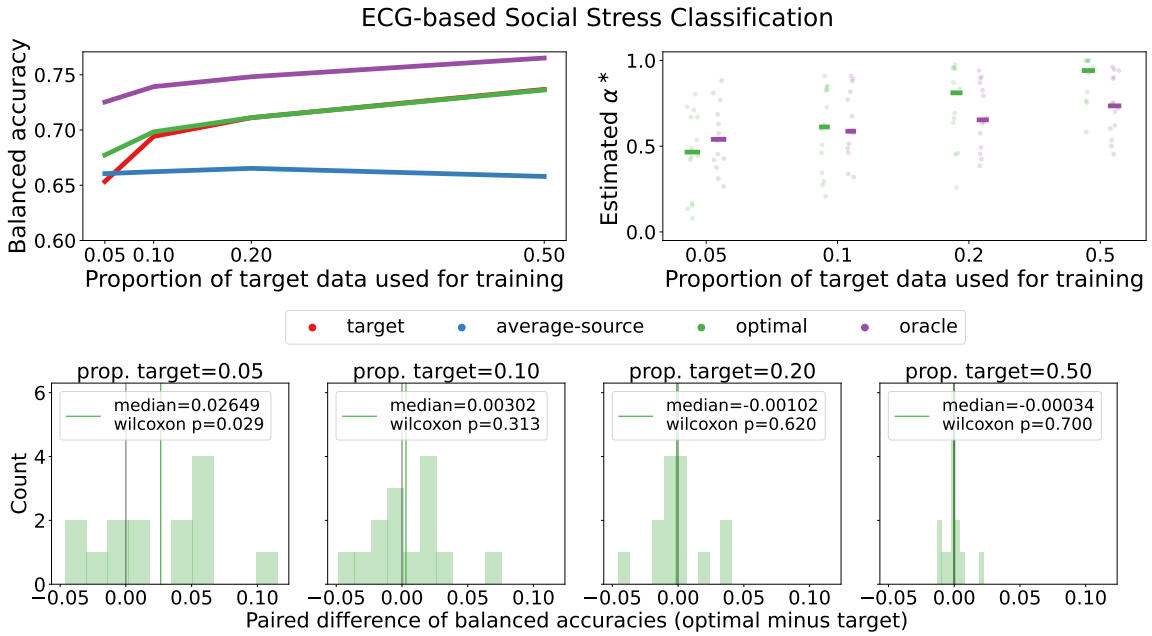


Figure A.11: Balanced accuracy and relevant convex coefficients (top) and relative performance of the optimal and target classifiers on a per-participant basis (bottom) for the Social Stress, ECG-based classification task.

The optimal classifier provided a measurable advantage only in the most data-scarce

condition. For $p = 0.05$, the optimal classifier outperformed the target classifier for the majority of participants, with a median improvement of 2.6% and a statistically significant Wilcoxon test (p-value = 0.029). For $p = 0.1$, 0.2, and 0.5, the paired differences were centered near zero and the Wilcoxon tests were not significant (p-values of 0.313, 0.620, and 0.700, respectively). The average-source classifier was never preferred over the target classifier at any data proportion, consistent with the very low empirical $\kappa \approx 1.5$: the source projection vectors were too dispersed to provide useful bias. This dataset represents a boundary case for the method, as the task similarity is low enough that the source data helps only when the target data is extremely scarce, and the advantage vanishes as soon as a modest amount of target data is available.

A.8 Derivation of the multivariate FLD risk

Suppose the target distribution is given by $P = \pi_0 P_0 + \pi_1 P_1$ where π_i is the prior probability and P_i is the class conditional density of the i-th class. The generative model in the main text specifies that $P_i = \mathcal{N}_d\left((-1)^{i+1}\nu, \Sigma\right)$. For simplicity, we only consider the case where $\pi_0 = \pi_1 = \frac{1}{2}$, but we note that the analysis can be easily extended to unequal priors. Under the 0-1 loss, the risk of an FLD hypothesis $\hat{h}(x) = \mathbb{1}\{\hat{\omega}^\top x > 0\}$ w.r.t to the target distribution P is given by,

$$
\begin{aligned}
R(h) &= \mathbb{P}_{X\sim P}\left[h(X) \neq Y\right] \\
&= \frac{1}{2}\mathbb{P}_{X\sim P_0}\left[\hat{\omega}^\top X > 0\right] + \frac{1}{2}\mathbb{P}_{X\sim P_1}\left[\hat{\omega}^\top X < 0\right] \\
&= \frac{1}{2} - \frac{1}{2}\mathbb{P}_{X\sim P_0}\left[\hat{\omega}^\top X < 0\right] + \frac{1}{2}\mathbb{P}_{X\sim P_1}\left[\hat{\omega}^\top X < 0\right]
\end{aligned}
$$

Since $\hat{\omega}^\top X \sim \mathcal{N}_1\left(\hat{\omega}^\top \mathbb{E}[X], \hat{\omega}^\top \Sigma\, \hat{\omega}\right)$, we have

$$R(h) = \frac{1}{2} - \frac{1}{2}\mathbb{P}\left[Z < \frac{\hat{\omega}^\top \nu}{\sqrt{\hat{\omega}^\top \Sigma\, \hat{\omega}}}\right] + \frac{1}{2}\mathbb{P}\left[Z < \frac{-\hat{\omega}^\top \nu}{\sqrt{\hat{\omega}^\top \Sigma\, \hat{\omega}}}\right],$$

where Z is a standard normal random variable. Therefore,

$$R(h) = \frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\hat{\omega}^\top \nu}{\sqrt{\hat{\omega}^\top \Sigma\, \hat{\omega}}}\right) + \frac{1}{2}\Phi\left(\frac{-\hat{\omega}^\top \nu}{\sqrt{\hat{\omega}^\top \Sigma\, \hat{\omega}}}\right).$$

Using the fact that $\Phi(-x) = 1 - \Phi(x)$, we arrive at the desired expression:

$$R(h) = \Phi\left(\frac{-\hat{\omega}^\top \nu}{\sqrt{\hat{\omega}^\top \Sigma\, \hat{\omega}}}\right).$$

Appendix B

Supplementary Materials for Chapter 3

B.1 Proof of Theorem 3.4.1

We prove the theorem in two stages. First, Lemma B.1.1 constructs a reference hypothesis sequence whose conditional expected prospective loss converges almost surely to Bayes risk. Then the main argument shows that prospective ERM does at least as well as this reference.

Throughout, we use the shorthand

$$e_m(h) \;\equiv\; \frac{1}{m}\sum_{s=1}^{m} \ell(s,\, h_s(X_s),\, Y_s)\,.$$

B.1.1 Unified subsequence construction

We begin by fixing a single increasing sequence of indices $\{i_t\}_{t=1}^{\infty}$ with $i_t \le t$ that will be used throughout the proof. Choosing such a sequence is possible because $\gamma_t \to 0$: we may extract a subsequence along which γ decays geometrically. Specifically, choose $\{i_t\}$ so that

$$\sum_{t=1}^{\infty} \sqrt{\gamma_{i_t}} \;<\; \infty, \tag{B.1}$$

and simultaneously i_t grows fast enough that the consistency condition Eq. (3.1) applied to $\mathcal{H}_{i_t}$ gives approximation errors that are summable. Crucially, $\{i_t\}$ depends

only on the sequence $\{\gamma_t\}$ and the approximation errors furnished by condition (i), not on the data Z. This is the non-adaptive choice asserted by the theorem.

B.1.2 Lemma: existence of a near-Bayes reference sequence

Lemma B.1.1. Let Z be a stochastic process and let $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots$ be an increasing sequence of hypothesis classes satisfying the consistency condition in Eq. (3.1). Let $\{i_t\}$ and $\{u_{i_t}\}$ be as above. Then there exists a sequence $\{h^{(t)}\}_{t=1}^{\infty}$ of random variables with $h^{(t)} \in \mathcal{F}_t = \sigma(Z_{\leq t})$ and $h^{(t)}$ $\mathcal{H}_{i_t}$-valued, such that

$$\mathbb{E}\left[\sup_{u_{i_t} \leq m < \infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_t\right] - R_t^* \;\longrightarrow\; 0 \quad \text{almost surely.} \tag{B.2}$$

Proof. **Step 1: Subsequence with small expected error.** By the consistency condition Eq. (3.1) applied to $\mathcal{H}_{i_t}$, we can choose for each t a hypothesis $h^{(t)} \in \sigma(Z_{\leq t})$ with $h^{(t)}$ $\mathcal{H}_{i_t}$-valued such that

$$\mathbb{E}\left[R_{i_t}\left(h^{(t)}\right) - R_{i_t}^*\right] \;\leq\; c_t,$$

where $\{c_t\}$ is summable (guaranteed by the choice of $\{i_t\}$ above). Because the limsup is a decreasing limit of tail suprema, the Bounded Convergence Theorem applied inside the conditional expectation gives

$$\mathbb{E}\left[R_{i_t}\left(h^{(t)}\right) - R_{i_t}^*\right] = \lim_{j \to \infty} \mathbb{E}\left[\mathbb{E}\left[\sup_{u_j \leq m < \infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_{i_t}\right] - R_{i_t}^*\right]. \tag{B.3}$$

Hence for each t there exists an index j_t such that

$$\mathbb{E}\left[\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_{i_t}\right] - R^*_{i_t}\right] \;\le\; 2c_t. \tag{B.4}$$

Step 2: Borel-Cantelli argument. Since $\{c_t\}$ is summable, Markov's inequality gives

$$\mathbb{P}\left(\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_{i_t}\right] - R^*_{i_t} \;>\; \sqrt{c_t}\right) \;\le\; \frac{2c_t}{\sqrt{c_t}} \;=\; 2\sqrt{c_t},$$

and $\sum_t 2\sqrt{c_t} < \infty$ (since c_t is summable and hence eventually smaller than any geometric rate). By the Borel–Cantelli lemma, with probability one there exists a random (but almost-surely finite) time t_0 such that for all $t \ge t_0$,

$$\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_{i_t}\right] - R^*_{i_t} \;\le\; \sqrt{c_t}. \tag{B.5}$$

Step 3: Lifting to the full filtration via the tower property. We now move from conditioning on $\mathcal{F}_{i_t}$ to conditioning on $\mathcal{F}_t$. Since $i_t \le t$, we have $\mathcal{F}_{i_t} \subseteq \mathcal{F}_t$. Because $h^{(t)}$ is $\mathcal{F}_{i_t}$-measurable (by construction), the tower property of conditional expectations gives

$$\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_{i_t}\right] \;=\; \mathbb{E}\left[\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_t\right] \,\middle|\, \mathcal{F}_{i_t}\right]. \tag{B.6}$$

Since $\mathbb{E}[X \mid \mathcal{F}_{i_t}] \ge \alpha$ a.s. implies $\mathbb{E}[X \mid \mathcal{F}_t] \ge \alpha$ a.s. when $\mathcal{F}_{i_t} \subseteq \mathcal{F}_t$ (by Jensen applied to the identity function), combining equation B.5 and equation B.6 yields, for all $t \ge t_0$ almost surely,

$$\mathbb{E}\left[\sup_{u_{j_t}\le m<\infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_t\right] - R^*_{i_t} \;\le\; \sqrt{c_t}. \tag{B.7}$$

Finally, since j_t may be larger than i_t and u is non-decreasing, we have $u_{j_t} \geq u_{i_t}$, so $\sup_{u_{i_t} \leq m} \geq \sup_{u_{j_t} \leq m}$. Combining this monotonicity with equation B.7 and $R_t^* \leq R_{i_t}^*$ (Bayes risk is non-increasing), we obtain

$$\mathbb{E}\left[\sup_{u_{i_t} \leq m < \infty} e_m\left(h^{(t)}\right) \,\middle|\, \mathcal{F}_t\right] - R_t^* \;\leq\; \sqrt{c_t} \;\longrightarrow\; 0 \quad \text{a.s.}$$

This is exactly equation B.2, completing the proof of the lemma. □

B.1.3 Main argument

We now show that prospective ERM inherits the near-Bayes property of the reference sequence. Fix $Z \in \mathcal{Z}$.

Step 1: Almost-sure concentration bound. We apply the concentration condition equation 3.2 with the sequence $\{i_t\}$ constructed above. Because $i_t \leq t$, the empirical maximum satisfies the monotonicity relation

$$\max_{u_{i_t} \leq m \leq i_t} e_m(h) \;\leq\; \max_{u_{i_t} \leq m \leq t} e_m(h) \quad \text{for all } h \in \mathcal{H}_{i_t}. \tag{B.8}$$

Together with the concentration condition, this gives

$$\mathbb{E}\left[\max_{h \in \mathcal{H}_{i_t}} \left|\bar{\ell}_t(h, Z) - \max_{u_{i_t} \leq m \leq t} e_m(h)\right|\right] \;\leq\; \gamma_{i_t}.$$

Applying Markov's inequality and using equation B.1,

$$\sum_{t=1}^{\infty} \mathbb{P}\left(\max_{h \in \mathcal{H}_{i_t}} \left|\bar{\ell}_t(h, Z) - \max_{u_{i_t} \leq m \leq t} e_m(h)\right| > \sqrt{\gamma_{i_t}}\right) \;\leq\; \sum_{t=1}^{\infty} \sqrt{\gamma_{i_t}} \;<\; \infty.$$

By the Borel–Cantelli lemma, with probability one there exists a random finite time t_1 such that for all $t \geq t_1$ and all $h \in \mathcal{H}_{i_t}$ simultaneously,

$$\bar{\ell}_t(h, Z) \;\leq\; \max_{u_{i_t} \leq m \leq t} e_m(h) + \sqrt{\gamma_{i_t}}. \tag{B.9}$$

Step 2: ERM does at least as well as the reference. Let $\hat{h}^{(t)}$ denote the prospective ERM estimator and let $h^{(t)}$ be the reference sequence from Lemma B.1.1 (applied with hypothesis classes $\{\mathcal{H}_{i_t}\}$ and sequence $\{u_{i_t}\}$). By definition, $\hat{h}^{(t)}$ minimizes $\max_{u_{i_t} \leq m \leq t} e_m(\cdot)$ over $\mathcal{H}_{i_t}$, and $h^{(t)} \in \mathcal{H}_{i_t}$. Therefore,

$$\max_{u_{i_t} \leq m \leq t} e_m\Big(\hat{h}^{(t)}\Big) \;\leq\; \max_{u_{i_t} \leq m \leq t} e_m\Big(h^{(t)}\Big)\,. \tag{B.10}$$

Step 3: Chaining the bounds. Fix any $t \geq \max(t_0, t_1)$. Applying equation B.9 to $\hat{h}^{(t)}$, then using the ERM inequality equation B.10, and finally enlarging the finite maximum to a supremum over $[u_{i_t}, \infty)$, we obtain

$$\begin{aligned}
\bar{\ell}_t\Big(\hat{h}^{(t)}, Z\Big) \;&\leq\; \max_{u_{i_t} \leq m \leq t} e_m\Big(\hat{h}^{(t)}\Big) + \sqrt{\gamma_{i_t}} \\
&\leq\; \max_{u_{i_t} \leq m \leq t} e_m\Big(h^{(t)}\Big) + \sqrt{\gamma_{i_t}} \\
&\leq\; \sup_{u_{i_t} \leq m < \infty} e_m\Big(h^{(t)}\Big) + \sqrt{\gamma_{i_t}}.
\end{aligned} \tag{B.11}$$

Taking conditional expectations given $\mathcal{F}_t$ and using Lemma B.1.1,

$$\mathbb{E}\Big[\bar{\ell}_t\Big(\hat{h}^{(t)}, Z\Big) \;\Big|\; \mathcal{F}_t\Big] \;\leq\; \mathbb{E}\left[\sup_{u_{i_t} \leq m < \infty} e_m\Big(h^{(t)}\Big) \;\middle|\; \mathcal{F}_t\right] + \sqrt{\gamma_{i_t}}$$

$$= R_t^* + o(1) + \sqrt{\gamma_{i_t}} \longrightarrow R_t^* \quad \text{a.s.} \tag{B.12}$$

The left-hand side is exactly $R_t\big(\hat{h}^{(t)}\big)$, so

$$R_t\big(\hat{h}^{(t)}\big) - R_t^* \longrightarrow 0 \quad \text{almost surely.} \tag{B.13}$$

Conclusion. Since the prospective risk is bounded (the loss ℓ is bounded), the Bounded Convergence Theorem applied to equation B.13 gives

$$\mathbb{E}\Big[R_t\big(\hat{h}^{(t)}\big) - R_t^*\Big] \longrightarrow 0.$$

Markov's inequality then yields, for every $\epsilon > 0$,

$$\mathbb{P}\Big(\Big|R_t\big(\hat{h}^{(t)}\big) - R_t^*\Big| \geq \epsilon\Big) \leq \frac{1}{\epsilon}\,\mathbb{E}\Big[R_t\big(\hat{h}^{(t)}\big) - R_t^*\Big] \longrightarrow 0.$$

Since this holds for every $Z \in \mathcal{Z}$ and the sequence $\{i_t\}$ depends only on $\{\gamma_t\}$ (not on Z), the prospective ERM learner is a strong prospective learner for the family $\mathcal{Z}$. ■

B.2 Proof of Theorem 3.4.2

The construction follows Hanneke (2021, Section 4) closely. We proceed in two stages: first we handle a fixed finite hypothesis class $\mathcal{H}$, constructing a suitable sequence u_t; then we extend to a countable $\mathcal{H}$ by an exhaustion argument that simultaneously produces $\mathcal{H}_t$ and γ_t.

Stage 1: finite hypothesis class

Fix a finite class $\mathcal{H}$ of hypothesis sequences. We construct u_t such that for all $Z \in \mathcal{Z}$,

$$\lim_{t\to\infty} \mathbb{E}\left[\sup_{t'\geq t} \max_{h\in\mathcal{H}} \left|\bar{\ell}_t(h, Z) - \max_{u_t\leq m\leq t'} e_m(h)\right|\right] = 0. \tag{B.14}$$

For a data sequence $\mathbf{z} = \{z_s\}_{s=1}^{\infty}$ and a hypothesis sequence $h \in \mathcal{H}$, define

$$t_u^h(\mathbf{z}) = \min\left\{t \in \mathbb{N} : t \geq u,\ \forall\, t' \geq t,\ \sup_{u\leq m<\infty} e_m(h) \leq \max_{u\leq m\leq t'} e_m(h) + 2^{-u}\right\},$$

the earliest time after which the maximum partial average lies within 2^{-u} of the limsup of $e_m(h)$ from index u onward. Define further

$$u_t^h(\mathbf{z}) = \max\left\{u \in \{1, \ldots, t\} : t \geq t_u^h(\mathbf{z})\right\},$$
$$u_t^{\mathcal{H}}(\mathbf{z}) = \min_{h\in\mathcal{H}} u_t^h(\mathbf{z}),$$

where the minimum over $\mathcal{H}$ exists because $\mathcal{H}$ is finite.

For the stochastic process Z, define the $(1-\delta)$-quantile of $u_t^{\mathcal{H}}$ by

$$u_t^{\mathcal{H}}(\delta, Z) = \max\left\{u \in \{1, \ldots, t\} : \mathbb{P}_{\mathbf{z}\sim Z}\left(u_t^{\mathcal{H}}(\mathbf{z}) \geq u\right) \geq 1 - \delta\right\}.$$

Set

$$u_t(Z) = \max\left\{s \in \mathbb{N} \cup \{0\} : u_t^{\mathcal{H}}(2^{-s}, Z) \geq s\right\},$$

$$u_t = \min_{Z \in \mathcal{Z}} u_t(Z).$$

Since $\mathcal{Z}$ is finite, the minimum is attained. By construction, for all $Z \in \mathcal{Z}$,

$$\mathbb{P}\Big(u_t^{\mathcal{H}}(\mathbf{z}) \geq u_t\Big) \geq 1 - 2^{-u_t}, \tag{B.15}$$

and $u_t \to \infty$ as $t \to \infty$ (since $\mathcal{Z}$ is finite and each $u_t(Z) \to \infty$). Applying the Borel–Cantelli lemma to equation B.15 gives equation B.14 for all $Z \in \mathcal{Z}$.

Stage 2: extension to a countable hypothesis class

Now let $\mathcal{H}$ be countable. Let $\{\mathcal{H}_i\}_{i=1}^{\infty}$ be any sequence of non-empty finite sets of hypothesis sequences with $\bigcup_{i=1}^{\infty} \mathcal{H}_i = \mathcal{H}$, and let $\{\gamma_i\}_{i=1}^{\infty} \subset (0, \infty)$ with $\gamma_1 \geq 1$ be a target rate sequence.

By Stage 1, for each $i \in \mathbb{N}$ there exists a sequence $\{u_{i,t}\}_{t=1}^{\infty}$ with $u_{i,t} \to \infty$, $u_{i,t} < t$, and for all $Z \in \mathcal{Z}$,

$$\lim_{t \to \infty} \mathbb{E}\left[\sup_{t' \geq t} \max_{h \in \mathcal{H}_i} \left| \bar{\ell}_t(h, Z) - \max_{u_{i,t} \leq m \leq t'} e_m(h) \right|\right] = 0. \tag{B.16}$$

Define

$$j_t = \max\left\{ i \in \{1, \ldots, t\} : \forall\, i' \leq i,\ \sup_{t'' \geq t} \mathbb{E}\left[\sup_{t' \geq t''} \max_{h \in \mathcal{H}_{i'}} \left| \bar{\ell}_{t''}(h, Z) - \max_{u_{i',t''} \leq m \leq t'} e_m(h) \right|\right] \leq \gamma_{i'} \quad \forall\, Z \in \mathcal{Z} \right\},$$

the largest index i up to t for which all classes $\mathcal{H}_1, \ldots, \mathcal{H}_i$ have already achieved their

target approximation rate. By equation B.16, $j_t \to \infty$. Define the stopping times

$$t_i = \min\Big\{t \in \mathbb{N} : j_t \geq i,\ u_{i,t} > u_{i-1,t_{i-1}}\Big\},$$

with $t_0 = 0$ and $u_{0,0} = 0$ by convention. Set

$$i_t = \max\{i \in \mathbb{N} : t_i \leq t\}, \qquad u_t = u_{i_t,\, t_{i_t}}.$$

Since $j_t \to \infty$ and $u_{i,t} \to \infty$ for each i, we have $i_t \to \infty$ and $u_t \to \infty$. By construction, for all $Z \in \mathcal{Z}$,

$$\mathbb{E}\left[\max_{h \in \mathcal{H}_{i_t}} \left|\bar{\ell}_t(h, Z) - \max_{u_t \leq m \leq t} e_m(h)\right|\right] \leq \gamma_{i_t}.$$

Setting $\mathcal{H}_t := \mathcal{H}_{i_t}$ and $\gamma_t := \gamma_{i_t}$, the sequences $\mathcal{H}_t$, u_t, and γ_t satisfy condition (i) of Theorem 3.4.1 (since $\bigcup_t \mathcal{H}_t = \mathcal{H}$ and $\mathcal{H}$ is dense in the sense of the consistency assumption) and condition (ii) with rate $\gamma_t \to 0$. This completes the proof. ■

Appendix C

Supplementary Materials for Chapter 4

C.1 Derivation of the Bayes-optimal state sequence for the foraging environment

This section derives the optimal state sequence for the prospective foraging environment described in Sections 3.7 and 4.6. Because the environment has deterministic dynamics and a known, periodic reward structure, the Bayes-optimal policy can be computed analytically.

Environment. The state space is $\mathcal{S} = \{0, 1, \ldots, 6\}$. Two reward patches are located at positions $s_A = 1$ and $s_B = 5$. The agent's action space is $\mathcal{A} = \{-1, 0, +1\}$, corresponding to moving left, staying, or moving right, with the resulting state clipped to the boundaries of the track. The dynamics are deterministic: $s_{t+1} = \text{clip}(s_t + a_t, 0, 6)$.

Reward structure. Rewards alternate between the two patches with period $2N$. During the first half-period ($t \bmod 2N < N$), patch A is active and its reward decays exponentially:

$$r_t(s_A) = e^{-\lambda(t \bmod r)}, \qquad r_t(s_B) = 0,$$

where $\lambda > 0$ is the decay rate. During the second half-period ($t \bmod 2N \geq N$), the roles reverse:

$$r_t(s_A) = 0, \qquad r_t(s_B) = e^{-\lambda((t \bmod r))}.$$

Reward at all other positions is zero regardless of time.

Travel time. The distance between the two patches is $|s_A - s_B| = 4$ cells. Since the agent moves at most one cell per time step, the minimum travel time between patches is $\tau_{\text{travel}} = 4$ steps. During travel, the agent collects zero reward.

Structure of the optimal policy. The optimal policy maximizes the cumulative discounted reward $\sum_{t=0}^{\infty} \gamma^t r_t(s_t)$. Because the reward structure is periodic with known period $2N$ and the dynamics are deterministic, the problem reduces to optimizing the departure time within each half-period.

Let t_{dep} denote the time step (relative to the start of the current half-period) at which the agent departs the active patch. If the agent departs at step t_{dep}, it collects reward $\sum_{k=0}^{t_{\text{dep}}-1} \gamma^k e^{-\lambda k}$ from the current patch and arrives at the alternative patch at step $t_{\text{dep}} + \tau_{\text{travel}}$. To arrive exactly when the alternative patch activates (at the start of the next half-period, step N), the agent must depart at $t^*_{\text{dep}} = N - \tau_{\text{travel}}$.

This departure time is optimal under the following argument. Each additional step spent at the active patch yields a marginal reward of $\gamma^{t_{\text{dep}}} e^{-\lambda t_{\text{dep}}}$, which is exponentially decreasing in t_{dep}. Each step of late departure delays arrival at the alternative patch by one step, causing the agent to miss the peak reward $e^0 = 1$ at the moment of activation and instead collect $e^{-\lambda}$ or worse. For any $\lambda > 0$ and γ sufficiently close to 1, the marginal cost of late arrival exceeds the marginal benefit of staying once t_{dep} is

large enough, and the crossover occurs at $t^*_{\text{dep}} = N - \tau_{\text{travel}}$, the latest departure time that permits on-time arrival.

The optimal state sequence. The Bayes-optimal state sequence is therefore periodic with period $2N$. In each half-period:

1. Steps 0 through $t^*_{\text{dep}} - 1$: remain at the active patch, collecting decaying reward.

2. Steps t^*_{dep} through $t^*_{\text{dep}} + \tau_{\text{travel}} - 1$: travel from the active patch to the alternative patch, collecting zero reward.

3. Steps $t^*_{\text{dep}} + \tau_{\text{travel}}$ through $r - 1$: the agent has arrived at the alternative patch; if $t^*_{\text{dep}} + \tau_{\text{travel}} = r$, arrival coincides exactly with activation and no idle time is wasted.

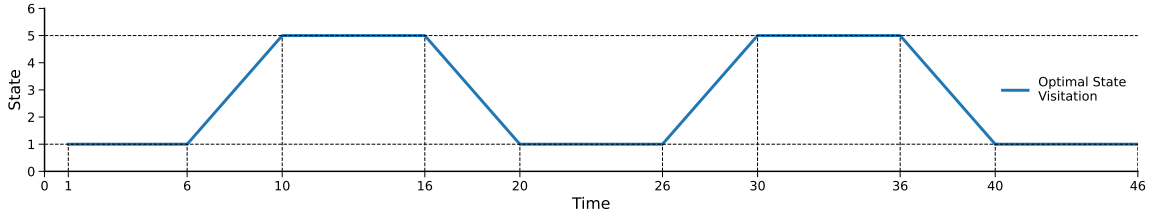


Figure C.1: The optimal state sequence for the foraging environment.

For the parameters used in the experiments ($N = 10$, $\tau_{\text{travel}} = 4$), the optimal departure time is $t^*_{\text{dep}} = 6$ steps into each half-period, and the agent arrives at the alternative patch at exactly step $N = 10$ — the moment of activation. As illustrated in Fig. C.1, the resulting state sequence alternates between four phases: harvest at s_A, travel from s_A to s_B, harvest at s_B, travel from s_B to s_A, with the transitions timed to coincide with the reward switches.

C.2 Fitted Q-Iteration

Fitted Q-Iteration (FQI) (Ernst et al., 2005; Munos and Szepesvari, 2008) is an offline reinforcement learning algorithm that approximates the state-action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ from a fixed dataset of interactions. Given a dataset $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{T}$ collected by a stochastic behavior policy, FQI alternates between two steps.

Target construction. At iteration k, bootstrap targets are formed using the current value estimate:

$$y_t^{(k)} = r_t + \gamma \max_{a' \in \mathcal{A}} Q_k(s_{t+1}, a').$$

Regression. The next iterate Q_{k+1} is obtained by fitting a function from a model class $\mathcal{F}$ to these targets via least squares:

$$Q_{k+1} = \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{t=1}^{T} \Big(f(s_t, a_t) - y_t^{(k)} \Big)^2.$$

After convergence, the learned $\hat{Q}$ induces a stationary greedy policy:

$$\pi(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \hat{Q}(s, a).$$

Time-Aware FQI. Standard FQI assumes a stationary environment and produces a policy that depends only on the current state. To handle non-stationary settings, we introduce a *time-aware* variant in which the Q-function is conditioned on the time index t, yielding $Q : \mathcal{S} \times \mathcal{A} \times \mathbb{N} \to \mathbb{R}$. The two steps of FQI are modified accordingly.

Target construction:

$$y_t^{(k)} = r_t + \gamma \max_{a' \in \mathcal{A}} Q_k(s_{t+1}, a', t).$$

Regression:

$$Q_{k+1} = \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{t=1}^{T} \Big(f(s_t, a_t, t) - y_t^{(k)} \Big)^2.$$

The resulting policy is *dynamic*, adapting its action selection to both state and time:

$$\pi(s, t) = \operatorname*{argmax}_{a \in \mathcal{A}} \hat{Q}(s, a, t).$$

This contrasts with the stationary policy produced by standard FQI, which cannot account for temporal shifts in the reward or transition structure.

In our experiments, we instantiate the function class $\mathcal{F}$ as a Random Forest with 1,000 trees, which provides a non-parametric regressor well-suited to capturing complex, non-linear value functions. The pseudocode for the online variant of FQI is given in Algorithm 1.

C.3 Soft Actor-Critic

Soft Actor-Critic (SAC) (Haarnoja, Zhou, Abbeel, et al., 2018; Haarnoja, Zhou, Hartikainen, et al., 2018) is an off-policy, model-free algorithm that augments the standard reinforcement learning objective with an entropy regularization term. Rather than maximizing cumulative return alone, the agent optimizes the entropy-regularized

Algorithm 1 Online Fitted Q-Iteration with ϵ-greedy exploration

Initialize foraging environment `env`, experience replay buffer $\mathcal{D}$, warmup size W, update interval I, initial exploration rate ϵ, action space $\mathcal{A}$
$\hat{Q} \leftarrow$ None
for $t = 1, 2, \ldots, T$
    $s_t \leftarrow$ `env`.get_current_state()
    Sample $u \sim \text{Uniform}[0, 1]$
    if $\hat{Q} =$ None **or** $u < \epsilon_t$, where $\epsilon_t = \max(0.01,\ \epsilon \cdot 0.999^t)$
        $a_t \leftarrow \text{UniformRandom}(\mathcal{A})$ ▷ Explore: draw a random action
    else
        $a_t \leftarrow \text{argmax}_{a' \in \mathcal{A}} \hat{Q}(s_t, a', t)$ ▷ Exploit: act greedily w.r.t. $\hat{Q}$
    end if
    $(s_{t+1}, r_t) \leftarrow$ `env`.step(a_t) ▷ Execute action, observe next state and reward
    Append $(s_t, a_t, r_t, s_{t+1}, t)$ to $\mathcal{D}$
    if $|\mathcal{D}| \geq W$ **and** $t \bmod I = 0$ ▷ Refit once warmed up, every I steps
        $\hat{Q} \leftarrow$ `FQI`.fit$(\mathcal{D})$
    end if
end for

objective,

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \Big(r_t + \alpha \, H\big(\pi(\cdot \mid s_t)\big) \Big) \right],$$

where $\gamma \in (0, 1]$ is the discount factor, $H(\cdot)$ denotes the Shannon entropy of the policy, and $\alpha > 0$ is a temperature parameter that governs the relative weight of the entropy bonus. Intuitively, the entropy term rewards the agent for maintaining a diverse action distribution, which encourages sustained exploration and guards against premature convergence to suboptimal deterministic policies. In practice, α can either be fixed or tuned automatically by treating it as a Lagrange multiplier subject to a minimum entropy constraint (Haarnoja, Zhou, Hartikainen, et al., 2018).

Algorithm 2 Online Soft Actor-Critic

Initialize foraging environment `env`, replay buffer $\mathcal{D}$, policy network π_θ, critic networks Q_{ϕ_1}, Q_{ϕ_2}, batch size B, Polyak coefficient ρ, temperature α
$\phi_{\text{targ},i} \leftarrow \phi_i$ for $i = 1, 2$ ▷ Initialize target critics
for $t = 1, 2, \ldots, T$
    $s_t \leftarrow$ `env`.get_current_state()
    $a_t \sim \pi_\theta(\cdot \mid s_t, t)$ ▷ Sample action from current policy
    $(s_{t+1}, r_t) \leftarrow$ `env`.step(a_t)
    Append $(s_t, a_t, r_t, s_{t+1}, t)$ to $\mathcal{D}$
    if $|\mathcal{D}| \geq B$
        Sample minibatch $\mathcal{B} \sim \mathcal{D}$ of size B
        $a' \sim \pi_\theta(\cdot \mid s_{t+1}, t{+}1)$ ▷ Next action for backup
        Compute critic targets:

$$y_t = r_t + \gamma\Big(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s_{t+1}, a', t{+}1) - \alpha \log \pi_\theta(a' \mid s_{t+1}, t{+}1)\Big)$$

        Update critics via gradient descent on, for $i = 1, 2$:

$$\nabla_{\phi_i} \frac{1}{B} \sum_{\mathcal{B}} \Big(Q_{\phi_i}(s_t, a_t, t) - y_t\Big)^2$$

        Update policy via gradient ascent on:

$$\nabla_\theta \frac{1}{B} \sum_{\mathcal{B}} \Big(\min_{i=1,2} Q_{\phi_i}(s_t, a_\theta, t) - \alpha \log \pi_\theta(a_\theta \mid s_t, t)\Big), \quad a_\theta \sim \pi_\theta(\cdot \mid s_t, t)$$

        Polyak-average target networks, for $i = 1, 2$:

$$\phi_{\text{targ},i} \leftarrow \rho\, \phi_{\text{targ},i} + (1 - \rho)\, \phi_i$$

    end if
end for

SAC maintains three learned components: a stochastic policy π_ϕ, two soft Q-functions $Q_{\theta_1}, Q_{\theta_2}$ (used to mitigate overestimation bias), and corresponding target networks $Q_{\bar{\theta}_1}, Q_{\bar{\theta}_2}$ updated via Polyak averaging. Taking the minimum over both critics when forming regression targets has been shown to substantially reduce the value overesti-

mation that destabilizes training (Haarnoja, Zhou, Abbeel, et al., 2018).

Time-Aware SAC. Analogously to the time-aware extension of FQI described in Section C.2, we construct a time-aware variant of SAC by conditioning the policy, critics, and target critics on the current time step t. Concretely, $\pi_\phi(a \mid s, t)$, $Q_{\theta_i}(s, a, t)$, and $Q_{\bar{\theta}_i}(s, a, t)$ each receive t as an additional input, enabling the agent to adapt its behavior to non-stationary dynamics or reward structures. The pseudocode for online SAC is given in Algorithm 2.

C.4 Proximal Policy Optimization

Proximal Policy Optimization (PPO) (Schulman et al., 2017) is an on-policy reinforcement learning algorithm that seeks to improve an agent's policy while ensuring that updates do not deviate too far from the current behavior.

While standard PPO is typically applied to episodic tasks with frequent environment resets, our implementation is adapted for a single-episode, reset-free regime. In this setting, the agent must continuously learn and adapt over a single, indefinite trajectory. We introduce several key modifications to the original framework:

Time-aware policy and value networks To handle the non-stationary nature of the foraging task, we make both the policy (π_θ) and critic (V_ϕ) functions of time t.

Infinite-Horizon Bootstrapping: In the absence of terminal states, we modify the Generalized Advantage Estimation (GAE). The advantage $\hat{A}_t$ is calculated by bootstrapping from the critic's value estimate of the next state for every transition,

treating the simulation as a continuous stream of experience rather than a series of independent episodes.

Online transition management The Rollout Buffer is utilized to store a fixed window of transitions. Updates are performed at regular intervals (every 512 steps), allowing for online policy refinement without the need for a global reset.

The online training loop of our PPO learner is outlined in Algorithm 3.

Algorithm 3 Online PPO for single-episode reset-free tasks

Initialize: Policy π_θ, Value network V_ϕ, Rollout Buffer $\mathcal{B}$
Set parameters: Clip ϵ, GAE parameters γ, λ, update interval N, epochs K
for $t = 0, 1, 2, \ldots, T$
    $s_t \leftarrow$ `env`.get_current_state()
    $a_t \sim \pi_\theta(\cdot \mid s_t, t)$
    $(s_{t+1}, r_t) \leftarrow$ `env`.step(a_t)
    Store $(s_t, a_t, t, r_t, \log \pi_\theta(a_t|s_t, t), V_\phi(s_t, t))$ in $\mathcal{B}$
    if $(t+1) \bmod N = 0$
        Bootstrap next value: $\hat{V}_{next} = V_\phi(s_{t+1}, t)$
        Estimate the advantages $\hat{A}_i$ using GAE
        Compute targets $R_i = \hat{A}_i + V_\phi(s_i, E_i)$ and normalize $\hat{A}$
        for $k = 1, \ldots, K$
            Calculate ratio $r_i(\theta) = \exp(\log \pi_\theta(a_i|s_i, t) - \log \pi_{old}(a_i|s_i, t))$
            $L_{clip} = \mathbb{E}_i[\min(r_i(\theta)\hat{A}_i, \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i)]$
            $L_v = \mathbb{E}_i[(V_\phi(s_i, t) - R_i)^2]$
            $L_{ent} = \mathbb{E}_i[\text{Entropy}(\pi_\theta(\cdot|s_i, t))]$
            Update θ, ϕ via Adam $\nabla_{\theta,\phi}(L^{clip} - c_1 L^v + c_2 L^{ent})$
        end for
        Clear buffer $\mathcal{B}$
    end if
end for

C.5 Pseudocode for the PLC Algorithm

Algorithm 4 Prospective Learner with Control (PLuC)

Initialize: Instantaneous loss model $g_i(s_t, t\,;\theta)$, Cumulative loss model $g_i(s_t, t\,;\varphi)$, Planning horizon H, Discount factor γ, Replay buffer $\mathcal{D}$, Initial position s_0

▷ `Warm-up phase`

for $t = 0, 1, 2, \ldots, T_{\text{warmup}} - 1$

  Randomly sample and move to the next allowed position s_t

  Observe the instantaneous reward r_t and store (s_t, a_t, t, r_t) in $\mathcal{D}$

end for

▷ `Online learning and planning phase`

for $t = T_{\text{warmup}} \ldots, T_{\text{terminal}}$

  ▷ `Update the models`

  Calculate cumulative rewards $\sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$

  Update g_i to minimize MSE on observed instantaneous rewards

  Update g_c to minimize MSE on calculated cumulative rewards

  ▷ `Selecting next action`

  Identify all possible action sequences $\boldsymbol{a}_{t:t+H}$

  for each sequence $\boldsymbol{a}_{t:t+H}$

    Obtain the resultant state sequence $\boldsymbol{s}_{t:t+H}$ by taking the actions $\boldsymbol{a}_{t:t+H}$

    Calculate the finite horizon loss $Q_{\text{finite}} = \sum_{h=1}^{H} \gamma^{h-1} g_i(x_{t+h}, t+h)$

    Calculate the terminal loss $Q_{\text{terminal}} = \gamma^H \cdot g_c(s_{t+H}, t+H)$

    Total sequence loss $Q(\boldsymbol{a}_{t:t+H}) = Q_{\text{finite}} + Q_{\text{terminal}}$

  end for

  ▷ `Learner-environment interaction`

  Execute the first action a_t^* of $\text{argmax}_{\boldsymbol{a}_{t:t+H}} Q(\boldsymbol{a}_{t:t+H})$

  Observe the instantaneous reward r_t and store (s_t, a_t^*, t, r_t) in $\mathcal{D}$

end for