

Learning When to Quit in Sales Conversations

Emaad Manzoor,^{1*} Eva Ascarza,² Oded Netzer³

¹ SC Johnson College of Business, Cornell University

² Stern School of Business, New York University

³ Columbia Business School, Columbia University

October 11, 2026

Abstract

Salespeople frequently face the dynamic qualification decision of whether to persist in a conversation or abandon it to pursue the next lead. Yet, little is known about whether these decisions are efficient or how firms can improve them. We study high-volume outbound sales, where leads are ample, time is scarce, and failure is common. We formalize dynamic qualification as an optimal stopping problem and develop a large language model (LLM)-based sequential decision agent — a *stopping agent* — that learns when to quit from historical conversations via offline reinforcement learning. When applied to calls from a large European telecommunications firm, our stopping agent reduces total call time by 67% while retaining 83% of observed sales. If the time saved is reallocated to new calls drawn from the same distribution, expected sales increase by up to 47%. Upon examining the linguistic cues associated with salespeople’s quitting decisions, we find that their decisions are disproportionately associated with a few salient expressions of prospect disinterest and appear to overlook early indicators of eventual failure. These patterns are consistent with cognitive bounds on salespeople’s ability to integrate and act on conversational information in real time. Our findings illustrate how offline reinforcement learning can be deployed to harness the natural language understanding capabilities of LLMs to improve the allocation of scarce salesperson time.

Keywords: salesforce management, telemarketing, reinforcement learning, agentic AI, text analysis, conversation data

*Emaad Manzoor (emaadmanzoor@cornell.edu) is the corresponding author. He is an Assistant Professor of Marketing at Cornell University. Eva Ascarza is the William Joyce Professor of Marketing at NYU Stern School of Business. Oded Netzer is the Arthur J. Samberg Professor of Business at Columbia Business School. We thank seminar participants at Northwestern University, Purdue University, Columbia University, the University of Texas at Austin, Carnegie Mellon University, Washington University in St. Louis, the University of Houston, and the University of Southern California, in addition to participants at the New Data for Consumer Insights Conference (especially our discussant Sanjog Misra), the Customer Intelligence Lab at Harvard, and the 2025 ISMS Marketing Science conference for their feedback. We also thank an anonymous firm for providing the data, and are grateful to Javier Serra de Paz and Daniel Domínguez Mosquera at the firm in particular. Open-source software is available at stoppingagents.com.

1 Introduction

Sales activities constitute over 5% of the U.S. GDP and employ 10% of the U.S. labor force [Misra, 2019]. Despite their economic significance, sales operations remain chronically time-inefficient: salespeople frequently spend valuable time on conversations that are unlikely to succeed [Dixon and McKenna, 2022]. This inefficiency is especially pronounced in high-volume settings like outbound call center sales — a \$97 billion global industry [Grand View, 2024] — where the supply of leads is ample, but salespeople’s time is limited and failure is common.

Much of the academic and managerial attention has centered on how to motivate selling effort. In contrast, little is known about the decision of whether and when to quit a sales call that is unlikely to succeed, which we refer to as the dynamic qualification problem. Given the prevalence of failure in high-volume sales, even marginal improvements in qualification efficiency can yield substantial time savings. In our empirical setting — an outbound sales campaign at a European telecommunications firm — failure is the norm: 94.5% of calls fail to end in a sale, and these unsuccessful calls (almost 3 minutes long on average) account for 82% of total call time. Every minute spent on a call that ends without a sale is a minute not spent on the next lead.

Yet, recognizing that a call is likely to fail is only part of the managerial challenge. The core decision is not just *how* to predict failure, but *when* to act on that prediction [Ascarza et al., 2021]. To that end, we formalize dynamic qualification as an optimal stopping problem, in which the algorithmic decision-maker must trade off the immediate benefit of quitting against the option value of continuing to gather information at each decision opportunity. Addressing this decision problem requires more than accurate predictions: it requires a principled framework for translating evolving conversational signals into timely actions that maximize cumulative payoffs.

To address this decision problem, we propose a *stopping agent*: an artificial intelligence agent built on a large language model (LLM) that silently observes an ongoing sales conversation and decides whether and when to terminate it in real time.¹ Although our stopping agent is built on an LLM, it never interacts with the prospect. Instead, it sequentially decides whether or not to quit the conversation based on the observed call transcript so as to maximize the firm’s expected cumulative payoff.

¹The stopping agent can also be configured to advise the salesperson to quit, instead of deciding directly.

Our key methodological contribution is an offline reinforcement learning procedure that enables LLMs to serve as truncation-only policy models for optimal stopping problems with textual states.² Our procedure builds on fitted Q -iteration (a classical batch Q -learning algorithm; Ernst et al., 2005). We retain its backward-induction structure but replace numerical regression with minimization of a weighted cross-entropy loss: the same loss used to pretrain LLMs. As a result, our procedure is implementable as standard supervised fine-tuning for both proprietary API-based and open-weight LLMs without architectural modifications (e.g., adding a regression head). This formulation allows the stopping policy to leverage the natural language understanding capabilities of LLMs to make forward-looking sequential decisions.

We provide theoretical justification for our procedure and show that it addresses the optimization difficulties associated with vanishing gradients in deep optimal stopping (DOS; Becker et al., 2019), a state-of-the-art approach for optimal stopping with deep neural networks. Further, our procedure can be used with non-LLM text-based policy models, such as logistic regression applied to a bag-of-words representation or embeddings of the conversation text.

We build a GPT-4.1 stopping agent and apply it to a dataset of 11,627 outbound sales calls collected over one month during a cross-selling campaign at our partner firm. When applied sequentially to held-out calls, our stopping agent with stopping opportunities 60 and 90 seconds into the call retains most sales (118 out of the 132 sales observed in the held-out calls) while reducing the total call time by 53%. Reallocating the time saved to new calls would increase expected sales by 41%. A variant of our stopping agent that is additionally allowed to quit 30 seconds into the call increases expected sales by 47% by reducing the total call time by 67%.

To understand the value of using LLMs as policy models, we estimate multiple variants of our proposed approach with policy models ranging from logistic regression to open-weight LLMs, as well as the API-based GPT-4.1 LLM. Expected sales gains increase monotonically, from a 2% loss with logistic regression to a 12% gain with the Gemma 3 270M open-weight LLM. The proprietary GPT-4.1 LLM policy increases expected sales by nearly four times the gain of Gemma 3 270M. These results highlight the importance of stopping policy estimation procedures that accommodate frontier API-based LLMs as policy models.

²A truncation-only stopping policy either terminates a conversation or lets it continue unchanged. It cannot prolong the conversation beyond when it would have ended otherwise, alter what the salesperson says, or otherwise change the salesperson’s behavior. The optimality of the stopping policy is defined within this restricted policy class.

We compare our approach with deep optimal stopping (DOS) [Becker et al., 2019] and deep recurrent optimal stopping (DROS) [Venkata and Bhattacharyya, 2023], two state-of-the-art offline reinforcement learning methods for optimal stopping using feedforward and recurrent neural network policy models, respectively. We also compare our approach with the classical fitted Q -iteration algorithm that estimates numerical continuation values via regression. In our empirical setting, our GPT-4.1 stopping agent consistently outperforms these reinforcement learning benchmarks. It also outperforms a fixed-deadline policy, policies based on call-outcome classifiers, and a sequential ensemble of these classifiers (with the latter two using the same GPT-4.1 model), highlighting the value of explicitly optimizing expected cumulative payoff.

We then assess whether our stopping agent generalizes beyond the campaign on which it was trained and whether it can generate decisions quickly enough for real-time deployment. Without retraining, we apply our stopping agent to a separate sales campaign conducted six months later, involving a different subscriber population and largely different salespeople. The GPT-4.1 stopping agent continues to improve expected sales, yielding gains of 15.7% with stopping opportunities at $t \in \{60, 90\}$ and 16.4% with stopping opportunities at $t \in \{30, 60, 90\}$. We also assess whether the decision latency of our stopping agents permits real-time use during sales calls. The median decision latency is 0.67 seconds for the GPT-4.1 agent and 26 milliseconds for the locally run Gemma 3 270M agent, both well below the 30-second interval between stopping opportunities.

The performance gap between salespeople and our stopping agent raises a managerial question: what distinguishes their quitting decisions? Examining the linguistic cues associated with these decisions, we find that salespeople’s quitting is strongly associated with a small set of salient phrases but less strongly associated with early indicators of eventual call failure. The phrase most predictive of salespeople quitting by far — “*no me interesa*” (“I’m not interested”) — rarely appears early in the call, suggesting that waiting for this salient expression of the prospect’s disinterest may delay qualification. In contrast, when our stopping agent quits earlier than salespeople, its decisions are associated with subtler indicators of a lack of conversational progress. These cues also vary as the conversation unfolds: early quitting decisions are associated with whether the salesperson is speaking to the right person, later decisions with the prospect’s interest, and still later decisions with whether the prospect already has alternatives to the salesperson’s offering.

Salespeople’s tendency to persist on calls likely to fail may reflect a low perceived opportunity cost of time. To examine this explanation, we study how salespeople prioritize which calls to quit near the end of their shift, when time is more constrained. Calls made near the end of the shift are shorter on average than those made earlier. However, we find that salespeople shorten calls throughout the predicted failure risk distribution, rather than concentrating the reduction among calls with higher predicted failure risk. While this analysis is descriptive, our findings suggest that pressure to quit earlier may not lead salespeople to quit the right calls, potentially reflecting difficulty predicting which calls are likely to fail.

Our research makes three key contributions, spanning substantive, methodological, and managerial dimensions. Substantively, we introduce dynamic qualification as an economically consequential sequential resource-allocation problem and show that optimizing quitting decisions can improve sales productivity by reallocating selling time through earlier disqualification.

Beyond documenting these gains, our analyses provide new insights into salesperson decision-making, showing that salespeople’s quitting decisions are strongly associated with a relatively limited set of salient, often late-arriving expressions, while early linguistic indicators of eventual call failure are less strongly associated with quitting. Together, these findings establish dynamic qualification as an important and previously understudied aspect of sales management.

Methodologically, our proposed offline reinforcement learning procedure broadens the class of policy models available for optimal stopping to include frontier LLMs and leverages their natural language understanding capabilities. In addition to being practically implementable as standard fine-tuning for both proprietary API-based and open-weight LLMs, our procedure is theoretically justified. As LLMs improve and expand to other modalities (such as audio and video), stopping agents trained using our procedure can improve alongside them and leverage data from these additional modalities. Our approach also “scales down” to simpler text-based policy models, such as logistic regression on bag-of-words representations or text embeddings, in settings where computational resources or training data are limited.

Managerially, we provide a practical approach to improve the efficiency of high-volume outbound sales campaigns. Our stopping agent can be implemented using readily available historical call transcripts, and our decision latency results demonstrate the computational feasibility of generating stopping decisions in real time, making deployment feasible in many real-world settings. In

our application, applying our stopping agent to held-out conversations saves time while retaining most observed sales; assuming that saved time can be reallocated to new calls drawn from the same distribution, these savings translate into substantial increases in expected sales. More broadly, our findings suggest that algorithmic decision support can assist salespeople with a cognitively demanding real-time allocation problem.

The rest of our paper is organized as follows. Section 2 reviews related literature. Section 3 presents our problem definition and proposed approach. Section 4 applies our method to call center sales and evaluates its performance. Section 5 explores systematic patterns in salespeople’s quitting decisions. Section 6 concludes, discusses limitations, and proposes directions for future research.

2 Related Work

This paper studies the dynamic screening decision of whether and when a salesperson should disqualify a prospect and end the ongoing sales conversation. While qualification is recognized as a core component of the selling process [Misra, 2019], dynamic qualification decisions *within* sales conversations have received little attention.

Our work extends the empirical salesforce management literature, which has primarily focused on mechanisms to motivate selling effort [Misra and Nair, 2011; Chung et al., 2014; Daljord et al., 2016; Kim et al., 2019, 2022; Bommaraju et al., 2025; Ahearne et al., 2025] (we refer the reader to Misra [2019] for a comprehensive review). We broaden the scope of this literature by examining how selling effort can be optimally withheld through dynamic disqualification and reallocated to more promising prospects. Rather than designing compensation schemes to motivate optimal qualification decisions, we propose an algorithmic approach to assist salespeople directly.

Our work also complements the growing literature on applying artificial intelligence to improve salesperson decision-making. Recent work has developed algorithms for salesforce recruitment [Chakraborty et al., 2025], pricing [Karlinisky-Shichor and Netzer, 2024], salesperson–prospect matching [Hu et al., 2024], sales forecasting from CRM activity logs [Reeder III et al., 2024], and sales training using LLMs [Habel et al., 2025]. We extend this literature by moving beyond prediction to optimize sequential sales decisions during live customer interactions. More broadly, our work contributes to the literature on LLMs as collaborators [Arora et al., 2025].

Our work also relates to the economics and marketing literature on optimal stopping and dynamic discrete choice [Rust, 1987]. This literature studies sequential decisions in which agents repeatedly choose whether to stop or not, trading off the payoff from acting now against the option value of waiting. Applications include consumer and firm decisions such as DVD preorders [Hui et al., 2008], bidding in freelance marketplaces [Yoganarasimhan, 2013], and consumer search [Zwick et al., 2003; Branco et al., 2016; Guo, 2022]. More recently, Kang et al. [2025] leverage the equivalence between dynamic discrete choice models and inverse reinforcement learning to estimate reward functions, while Barzegary and Yoganarasimhan [2025] propose recursive state-space partitioning to address high-dimensional problems. Methodologically, this literature typically uses observed dynamic choices to recover the primitives governing decision-making and, in many cases, relies on optimality assumptions to rationalize observed behavior.

We build on the same sequential trade-off between stopping and continuing, but pursue a different objective. Specifically, we do not estimate structural primitives or assume that decision-makers behave optimally. Instead, we develop a prescriptive algorithm to improve potentially suboptimal stopping decisions, using historical calls as training data rather than as a dataset of optimal choices to be structurally rationalized. Our setting changes the role of dynamic choice models from explaining observed behavior to learning a policy that can improve upon it. We further provide evidence suggesting systematic inefficiencies in salespeople’s stopping decisions.

Our methodological approach builds on reinforcement learning and numerical methods for finite-horizon optimal stopping. A common feature of these approaches is backward induction: starting from the final time period, they recursively estimate the value of continuing relative to stopping at each earlier decision point. This structure underlies early regression-based backward-induction methods used in finance [Longstaff and Schwartz, 2001; Tsitsiklis and Van Roy, 2001], as well as subsequent approaches such as fitted Q -iteration [Ernst et al., 2005] and deep optimal stopping [Becker et al., 2019]. We retain the backward induction structure but adapt the estimation to accommodate LLMs whose standard training objective is cross-entropy loss minimization.

Specifically, we replace numerical regression with minimization of a weighted cross-entropy loss. Since this is the same loss used in LLM pretraining and supervised fine-tuning, our procedure allows pretrained LLMs to be used directly as policy models for optimal stopping, without introducing architectural modifications such as regression heads. This formulation enables optimal

stopping policies to leverage modern LLMs while preserving the standard fine-tuning loss using which both proprietary API-based and open-weight models are trained. As we demonstrate in Section 4, this adaptation is consequential in our setting: the resulting LLM-based stopping policy outperforms these alternative optimal-stopping methods.

Our work also relates to the literature that applies reinforcement learning to LLMs, including methods used to align or fine-tune LLM behavior such as Proximal Policy Optimization (PPO; Schulman et al., 2017) and Group Relative Policy Optimization (GRPO; Shao et al., 2024). In these approaches, the LLM generates a sequence of tokens as actions, receives rewards based on the generated sequence, and is updated to favor higher-reward sequences. Notably, the LLM’s actions determine the evolution of the state, since the next state is formed by appending generated tokens to the preceding context [Lambert, 2025, §6.3.2.2].

Our use of reinforcement learning with LLMs is different. The LLM does not generate token sequences that fully determine how the textual state evolves. Instead, it observes a conversation whose evolution is governed by the interaction between the prospect and the salesperson and chooses between stopping the conversation and allowing it to continue. Thus, our problem combines an LLM policy with externally governed state transitions, rather than using reinforcement learning to optimize the LLM when the next state is determined by its own generated tokens.

Finally, our work contributes to the behavioral economics literature that uses machine learning to evaluate the quality of human decisions [Kleinberg et al., 2018; Mullainathan and Obermeyer, 2022; Rambachan, 2024]. Existing studies examine one-shot screening decisions, such as whether to detain a defendant or administer a medical test, and assess deviations from optimal threshold rules. We extend this literature by analyzing deviations from optimal stopping behavior in a sequential decision environment.

3 Optimal Conversation Stopping Agents

We now develop a stopping agent that observes an evolving sales conversation and makes sequential decisions about whether or not to stop the conversation. We first formalize this dynamic qualification decision as a finite-horizon optimal stopping problem and characterize how the optimal policy can be estimated from historical conversations using offline reinforcement learning.

We then develop our policy estimation procedure, which builds on fitted Q -iteration but replaces continuation-value regression with weighted cross-entropy loss minimization. This enables standard fine-tuning of LLMs as policy models for optimal stopping with textual states.

3.1 Problem Definition

Since conversations evolve sequentially, stopping decisions are inherently dynamic. Each time period spent on a call generates information about whether it is likely to end in a sale, but also incurs an opportunity cost because that time could instead be allocated to another prospect. This trade-off gives rise to an optimal stopping problem: deciding whether to stop now or continue gathering information to inform future stopping decisions. We adapt the discrete-time finite-horizon optimal stopping formulation of Shiryaev [1978] to this setting.

3.1.1 States. In each time period $t \in \{1, \dots, T-1\}$ during an ongoing conversation, the state $s_t \in \mathcal{S}$ consists of the verbatim transcript of the conversation observed up to time period t . Let T denote a dummy terminal time period corresponding to the observed end of the conversation, so that s_T is the transcript of the entire conversation. States evolve according to the unknown transition dynamics $\mathcal{P}_t(s_{t+1}|s_t)$, governed by the ongoing interaction between the salesperson and the prospect.

3.1.2 Actions. Let $\pi_t^{\theta_t}$ denote the stopping policy at time period t , parameterized by θ_t . Specifically, the policy observes s_t and chooses an action $\pi_t^{\theta_t}(s_t) = a_t \in \{0, 1\}$, where $a_t = 1$ denotes stopping the conversation and $a_t = 0$ denotes continuing. If the policy never stops before $t = T$, the conversation evolves according to $\mathcal{P}_t(s_{t+1}|s_t)$ until it ends at T . Otherwise, the conversation is truncated at stopping time $\tau < T$. For notational convenience, we treat a policy that never stops as one that “stops” at T , so that $\tau \in \{1, \dots, T\}$ is always well-defined and $a_T \equiv 1$.

We refer to this class of policies as *truncation-only* policies because they may terminate a conversation early or allow it to continue unchanged, but cannot prolong a conversation, alter what is said, or otherwise change the salesperson’s behavior. This restriction reflects our managerial objective of improving early qualification decisions. More general policies (e.g., agents that coach salespeople or generate responses) are outside the scope of this paper.

As we develop our stopping agent in this section, we leave the policy model unspecified, allowing $\pi_t^{\theta_t}$ to represent any model that maps conversational text into a binary stopping decision. Later, in Section 4, we instantiate our approach using a range of policy models, from logistic regression to frontier LLMs.

3.1.3 Rewards. Let $g(\tau, s_\tau)$ denote the cumulative reward associated with stopping the conversation at time τ . The reward function is specified by the firm and encodes its managerial objective. For example, it may take the form:

$$g(\tau, s_\tau) = \begin{cases} -c(\text{truncated conversation length}) & \text{if } \tau < T \\ bY - c(\text{full conversation length}) & \text{if } \tau = T, \end{cases} \quad (1)$$

where $c(\cdot)$ denotes the opportunity cost of spending time on the conversation, b is the value of a completed sale, and $Y \in \{0, 1\}$ indicates whether a sale is observed.

This reward function captures the trade-off central to dynamic qualification: stopping early saves time but forgoes the possibility of completing a sale, whereas continuing preserves that possibility while incurring an opportunity cost. More generally, our methodology accommodates any reward function defined over stopping times and conversation outcomes.

A key feature of our problem is that the reward from stopping is immediately known, whereas the value of continuing is not. At any ongoing conversation state s_t , choosing $a_t = 1$ yields the known reward $g(t, s_t)$. In contrast, the value of continuing depends on the future conversation and the stopping time $\tau \in \{t + 1, \dots, T\}$ induced by subsequent stopping decisions. Thus, deciding whether to stop requires comparing a known reward with the expected value of continuing and potentially stopping later.

3.1.4 Stopping Policy Objective. Given a dataset $\mathcal{D} = \{(s_1^k, \dots, s_T^k, Y^k)\}_{k=1}^M$ of M historical sales calls and their outcomes, our objective is to estimate a stopping policy that maximizes the expected reward:

$$J(\theta_1, \dots, \theta_{T-1}) = \mathbb{E}[g(\tau, s_\tau)], \quad (2)$$

where τ denotes the stopping time induced by the policy, and the expectation is taken over the state dynamics $\mathcal{P}_t(s_{t+1} | s_t)$ and the induced stopping time τ .

The historical conversations in $\mathcal{D}$ were collected without a stopping agent and therefore contain the full conversation observed under the salesperson’s behavioral policy. From the perspective of our stopping problem, every non-terminal time period corresponds to continuation: if a historical conversation is observed at both t and $t + 1$, then $a_t = 0$. The observed end of the conversation defines its terminal period T , at which $a_T = 1$ by definition.³

Our estimation procedure uses these calls to estimate $\theta_1, \dots, \theta_{T-1}$ that maximize Equation (2). We next characterize the optimal policy when the state transition dynamics are known and then show how to estimate it from historical conversations when those dynamics are unknown.

3.2 Estimating the Optimal Stopping Policy

3.2.1 Bellman Recursion and the Need for Offline Reinforcement Learning. For tractable state spaces with known transition dynamics, the optimal stopping policy can be computed directly by backward induction [Bellman, 1966]. Let $V_t^*(s)$ denote the maximum expected reward obtainable from period t onward given state $s_t = s$ (i.e., the optimal value function). The optimal value functions satisfy Bellman’s optimality recursion:

$$V_t^*(s) = \max \{g(t, s), \mathbb{E}[V_{t+1}^*(s_{t+1}) | s_t = s, a_t = 0]\} \text{ for } t < T, \quad V_T^*(s) = g(T, s). \quad (3)$$

The optimal policy stops whenever the reward from stopping exceeds the expected value of continuing, $g(t, s_t) > \mathbb{E}[V_{t+1}^*(s_{t+1}) | s_t, a_t = 0]$.

This recursion clarifies the central estimation problem: the stopping reward $g(t, s_t)$ is known, but the continuation value is not. In our setting, the continuation value cannot be computed directly because the conversational state consists of high-dimensional text and the transition dynamics governing how conversations evolve are unknown. We must therefore estimate the stopping policy from historical conversations, turning this into an offline reinforcement learning problem.

³This does not imply that salespeople never choose to end conversations. Rather, the salesperson’s decision to end a historical conversation determines its observed terminal period T ; at every earlier time period, the conversation continued by definition.

3.2.2 Coverage and Concentrability Under Truncation-Only Policies. Estimating a policy via offline reinforcement learning generally requires sufficient *coverage* and *concentrability*. We now detail how the truncation-only policy class we consider relaxes these requirements.

Coverage. Coverage requires the behavioral policy that generated the historical data to choose, with positive probability, every action that the estimated policy may choose in every state it may reach [Sutton and Barto, 2018, Section 5.5]. In our setting, the behavioral policy chooses $a_t = 0$ at every non-terminal state, so the stopping action $a_t = 1$ is not observed at any $t < T$.

This lack of coverage for $a_t = 1$ does not prevent policy estimation because stopping immediately terminates the conversation and yields the known reward $g(t, s_t)$. Its value therefore need not be estimated. In contrast, a truncation-only policy leaves the conversation unchanged until the stopping agent terminates it. Every state at which the agent chooses to continue is also reached by the behavioral policy, which continues in that state. Thus, $a_t = 0$ is fully covered.

Concentrability. Concentrability requires a uniform finite bound on how much more likely a state-action pair can be under a policy than under the distribution supplying the historical data [Chen and Jiang, 2019, Assumption 1; Agarwal et al., 2019, Assumption 4.1]. Intuitively, this condition limits how frequently a policy visits state-action pairs that are rare in the historical data.

For stopping, $a_t = 1$, no finite bound exists. A truncation-only policy can stop at a reachable non-terminal state with positive probability, whereas the behavioral policy has zero probability of stopping there. The corresponding probability ratio is infinite. This does not prevent policy estimation because stopping terminates the conversation and yields the known reward $g(t, s_t)$; its value need not be estimated from historical conversations.

For continuation, $a_t = 0$, the probability ratio relative to the behavioral policy at the same time period is bounded by one. Until the stopping agent terminates the conversation, the conversation evolves in the same way as under the behavioral policy. This is because stopping earlier can only reduce the probability of reaching a state and continuing in that state.

Hence, the historical data-generating process provides the required support for estimating continuation values via offline reinforcement learning, while the stopping values are known. We next describe how FQI estimates these continuation values from historical conversations.

3.2.3 Fitted Q -Iteration (FQI). Fitted Q -iteration (FQI; Ernst et al., 2005) provides a natural starting point for estimating stopping policies via offline reinforcement learning. FQI approximates the Bellman recursion by recursively estimating continuation-value functions through regression. This backward-induction procedure has previously been used for finite-horizon optimal stopping problems, particularly in finance [Longstaff and Schwartz, 2001; Tsitsiklis and Van Roy, 2001]. Algorithm 1 shows finite-horizon FQI [Bagnell et al., 2023, Algorithm 10] applied to our setting.

Algorithm 1 Finite-Horizon Fitted Q -Iteration

Require: Historical conversation dataset $\mathcal{D} = \{(s_1^k, \dots, s_T^k, Y^k)\}_{k=1}^M$

- 1: Define $\hat{Q}_T^{\hat{\theta}_T}(s_T^k, 0) \leftarrow g(T, s_T^k)$ for $k = 1, \dots, M$ ▷ Base case
 - 2: **for** $t = T - 1, T - 2, \dots, 1$ **do**
 - 3: $y_t^k \leftarrow \max\{g(t + 1, s_{t+1}^k), \hat{Q}_{t+1}^{\hat{\theta}_{t+1}}(s_{t+1}^k, 0)\}$ for $k = 1, \dots, M$ ▷ Construct targets
 - 4: $\hat{\theta}_t \leftarrow \arg \min_{\theta_t} \frac{1}{M} \sum_{k=1}^M \left(\hat{Q}_t^{\theta_t}(s_t^k, 0) - y_t^k \right)^2$ ▷ Minimize regression loss
 - 5: **end for**
 - 6: **return** $\{\hat{\theta}_t\}_{t=1}^{T-1}$
-

Algorithm 1 takes the historical conversation dataset $\mathcal{D}$ as input. Since every conversation ends at T , the terminal continuation value is known and equals the terminal reward $g(T, s_T^k)$.

Starting from the terminal condition, FQI recursively estimates continuation values for periods $t = T - 1, \dots, 1$. In each period t , it uses the continuation-value function already available for $t + 1$ to construct a regression target for every historical conversation. This target is the larger of the reward from stopping at $t + 1$ and the estimated value of continuing at $t + 1$ (Line 3). The algorithm then fits the current continuation-value function $\hat{Q}_t^{\hat{\theta}_t}(s_t, 0)$ to these targets using regression (Line 4).⁴

Once the continuation-value functions have been estimated, the stopping policy is given by:

$$\hat{\pi}_t^{\hat{\theta}_t}(s_t) = \mathbb{1} \left[g(t, s_t) > \hat{Q}_t^{\hat{\theta}_t}(s_t, 0) \right] \forall t < T. \quad (4)$$

3.2.4 Adapting FQI to LLMs as Policy Models. Applying FQI in our setting therefore requires estimating an accurate mapping from textual states to continuation values, where each state is the full conversation transcript observed up to the current time period. While FQI does not preclude the use of textual states, the effectiveness of the estimated stopping policy depends on the

⁴Because $Q_t(s, 1) = g(t, s)$ is known, only $Q_t(s, 0)$ must be estimated.

policy model’s ability to understand natural language and map decision-relevant information in conversations to forward-looking stopping decisions.

In principle, one could apply FQI to conversation transcripts using any policy model capable of mapping text to a numerical continuation value. One approach is to first convert the transcript into a fixed numerical representation, such as bag-of-words features or text embeddings, and then estimate the policy via regression (as in Algorithm 1). This approach is straightforward, but its performance will suffer if the fixed representation does not fully capture the linguistic information that is predictive of the value of continuing the conversation.

Another approach is to use an LLM directly as the policy model. However, this approach introduces additional architectural and training decisions (e.g., attaching a regression head that predicts a scalar value and minimizing a regression loss), whose effects on performance have received limited systematic study [Lukasik et al., 2025] and may be detrimental. Moreover, this approach is applicable only to models whose architecture and training loss can be modified (e.g., open-weight LLMs such as Gemma 3). For proprietary API-based LLMs like GPT, users cannot attach a regression head or change the cross-entropy loss.

Thus, while FQI can accommodate textual states, applying it to conversational text presents two practical limitations. Using fixed numerical representations preserves the standard regression formulation of FQI but may sacrifice information contained in the full conversational text. Using LLMs directly requires adapting them to numerical continuation-value regression, which requires architectural modification for open-weight LLMs⁵ and is generally not supported by the standard fine-tuning interfaces of proprietary API-based models.

Our approach addresses these challenges by retaining FQI’s backward-induction logic while replacing numerical continuation-value regression with an estimation problem compatible with standard cross-entropy language-model fine-tuning. This reformulation preserves the fundamental logic of FQI while enabling the stopping policy to exploit rich pretrained LLMs through their native fine-tuning interfaces, including proprietary API-based LLMs.

3.2.5 Proposed Approach. Like FQI, our approach takes the historical conversation dataset $\mathcal{D}$ as input and estimates period-specific models recursively backward from the terminal period

⁵We evaluate this implementation empirically in Section 4.4.3 and find that FQI performs worse than our proposed approach when paired with transformer-based policy models, including open-weight LLMs.

T . Rather than estimating a continuation-value function, however, it estimates a stopping score $F_t^{\theta_t}(s_t) \in (0, 1)$ by minimizing a weighted cross-entropy loss.

The induced policy stops when the stopping score exceeds a period-specific threshold κ_t :

$$\hat{\pi}_t^{\theta_t}(s_t) = \mathbb{1} \left[\hat{F}_t^{\theta_t}(s_t) > \kappa_t \right]. \quad (5)$$

We now describe the construction of our loss function. In each period t , we compare the reward from stopping immediately with the reward from continuing and following the estimated downstream policy. Let τ_{t+1}^k denote the stopping time induced by the downstream policy on conversation k , which is known since the policy models for periods $t + 1, \dots, T - 1$ have been estimated (i.e., by induction). We define the resulting stopping advantage as:

$$\Delta_t^k = g(t, s_t^k) - g(\tau_{t+1}^k, s_{\tau_{t+1}^k}^k). \quad (6)$$

For conversation k , the sign of Δ_t^k identifies which action yields the larger realized reward for that conversation and period, while its magnitude measures how much better that action is. We define the binary target and observation weight as:

$$y_t^k = \mathbb{1}[\Delta_t^k > 0], \quad w_t^k = |\Delta_t^k|, \quad (7)$$

and estimate the parameters of the policy model by minimizing the following weighted binary cross-entropy loss in each time period:

$$\hat{\theta}_t = \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k \left[y_t^k \log F_t^{\theta_t}(s_t^k) + (1 - y_t^k) \log (1 - F_t^{\theta_t}(s_t^k)) \right]. \quad (8)$$

Equation (8) can be viewed as estimating a cost-sensitive binary classifier in each time period: y_t^k identifies the preferred action, while w_t^k determines the cost of taking the wrong action. Importantly, these targets are not observed salesperson decisions or realized call outcomes; they are recursively constructed labels indicating whether stopping or continuing yields the higher reward under the estimated downstream policy.

Algorithm 2 summarizes the complete estimation procedure. The key deviation from Algorithm 1 occurs in lines 4–8: instead of constructing numerical continuation-value targets and estimating them by regression, our proposed approach constructs binary targets and observation weights from realized reward differences and minimizes the weighted cross-entropy loss.

Algorithm 2 Proposed Approach

Require: Historical conversation dataset $\mathcal{D} = \{(s_1^k, \dots, s_T^k, Y^k)\}_{k=1}^M$

```

1: Set  $\hat{F}_T(s_T^k) \leftarrow 1$  for  $k = 1, \dots, M$  ▷ Base case
2: for  $t = T - 1, T - 2, \dots, 1$  do
3:   for  $k = 1, \dots, M$  do
4:      $\tau_{t+1}^k \leftarrow \min\{u \in \{t + 1, \dots, T\} : \hat{F}_u^{\hat{\theta}_u}(s_u^k) > \kappa_u\}$  ▷ Downstream stopping time
5:      $\Delta_t^k \leftarrow g(t, s_t^k) - g(\tau_{t+1}^k, s_{\tau_{t+1}^k}^k)$ 
6:      $y_t^k \leftarrow \mathbb{1}[\Delta_t^k > 0], w_t^k \leftarrow |\Delta_t^k|$  ▷ Binary target and weight
7:   end for
8:    $\hat{\theta}_t \leftarrow \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k \left[ y_t^k \log F_t^{\theta_t}(s_t^k) + (1 - y_t^k) \log(1 - F_t^{\theta_t}(s_t^k)) \right]$ 
9: end for
10: return  $\{\hat{\theta}_t\}_{t=1}^{T-1}$ 

```

Web Appendix A provides a theoretical justification for Algorithm 2. We show that, holding the estimated downstream policy fixed, thresholding the minimizer of our weighted cross-entropy objective recovers the expected-reward-maximizing current-period stopping decision. Applying this argument recursively backward implies that, with a sufficiently expressive policy class, the resulting policy can approximate the maximum expected cumulative reward arbitrarily closely.

3.2.6 Implementation with Standard LLM Fine-Tuning. Since our proposed approach reformulates each period-specific estimation problem as weighted cross-entropy loss minimization, it can be implemented using standard LLM fine-tuning. Standard LLM fine-tuning requires a training dataset of natural-language prompt–response pairs. Accordingly, we convert each conversational state s_t^k into a prompt containing the transcript observed through time t together with instructions asking whether the conversation should be terminated. Figure 1 illustrates this construction. The expected response simply verbalizes the binary target: “yes” when $y_t^k = 1$ and “no” otherwise.

The weighted cross-entropy loss additionally requires weighting each training example according to w_t^k . Since proprietary fine-tuning APIs typically do not permit observation-specific weights, we implement weighting through resampling. Specifically, for each time period t , prompt–response

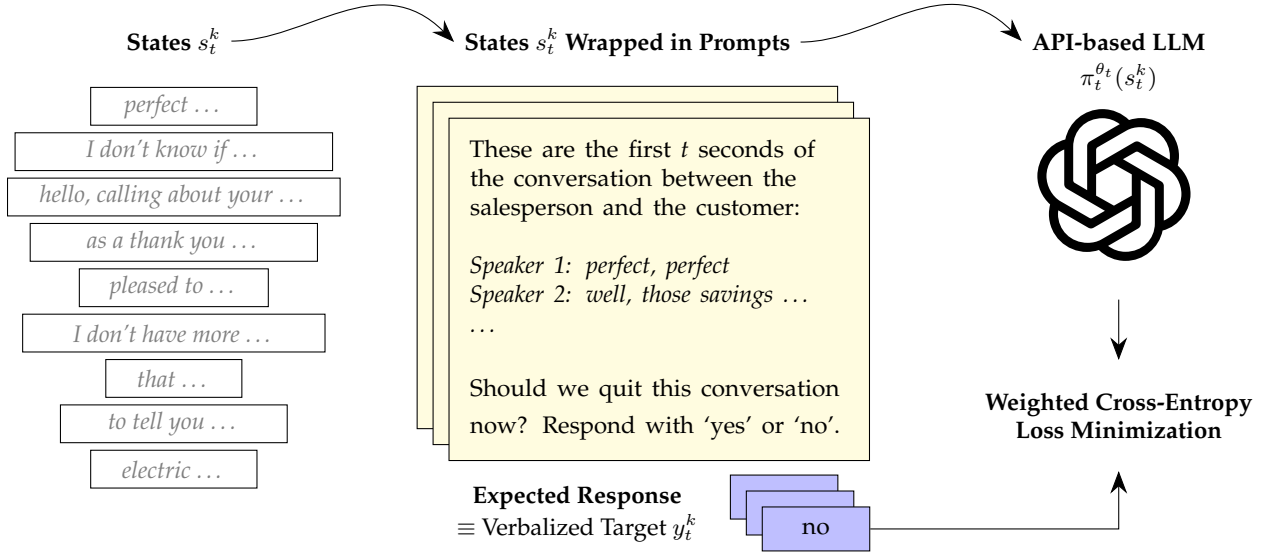


Figure 1: Implementation of Algorithm 2 using standard LLM fine-tuning. Each conversational state s_t^k is converted into a natural-language prompt and paired with the corresponding verbalized binary target ($y_t^k = \text{"yes" or "no"}$).

pairs are sampled with probabilities proportional to w_t^k , so that observations with larger weights appear more frequently during training. This resampling procedure is equivalent, in expectation, to minimizing the weighted cross-entropy loss in Equation (8).

3.2.7 Relationship to Deep Optimal Stopping (DOS). Having established our proposed approach, we next relate it to deep optimal stopping (DOS; Becker et al., 2019), a closely related approach. Both our approach and DOS proceed backward from the terminal period and compare the reward from stopping immediately with the reward from continuing and following the estimated downstream policy. However, the approaches differ in how they estimate the current-period stopping score $F_t^{\theta_t}(s_t)$ from these reward comparisons.

DOS estimates θ_t , for each $t < T$, by minimizing the following [Becker et al., 2019, Equation 15]:

$$-\frac{1}{M} \sum_{k=1}^M \left[g(t, s_t^k) F_t^{\theta_t}(s_t^k) + g(\tau_{t+1}^k, s_{\tau_{t+1}}^k) (1 - F_t^{\theta_t}(s_t^k)) \right]. \quad (9)$$

Web Appendix B characterizes how this objective differs from ours. We show that the DOS objective is equivalent to minimizing the weighted mean absolute error (MAE) with our targets

and observation weights, and hence inherits the vanishing-gradient problem associated with MAE [Zhang and Sabuncu, 2018]. Our loss, in contrast, does not suffer from vanishing gradients.

Web Appendix B also provides simulation evidence of the resulting finite-sample optimization difficulties, using the Bermudan max-call option-pricing environment studied by Becker et al. [2019]. Holding the policy model and training setup fixed, we compare the performance of our weighted cross-entropy loss with the DOS loss. We show that our approach outperforms DOS, particularly when the training sample is limited and the policy model is large, despite differing from DOS only in the loss function.

3.3 Putting It All Together: Stopping Agents

The preceding subsections formalize dynamic qualification as an optimal stopping problem and develop our key methodological contribution: a policy estimation procedure that preserves the backward-induction logic of FQI while replacing continuation-value regression with weighted cross-entropy loss minimization. This reformulation enables LLMs to serve as policy models for optimal stopping problems with textual states using their standard fine-tuning loss. Our procedure is compatible with both proprietary API-based LLMs and open-weight LLMs, as well as with simpler non-LLM policies such as logistic regression applied to text embeddings.

The resulting stopping agent is estimated offline on historical conversations and operates sequentially during live conversations. It does not communicate with the prospect or alter the salesperson’s behavior. Instead, it observes the transcript as the conversation unfolds and decides whether the conversation should be terminated so that the salesperson can move to the next lead. This design directly addresses the managerial problem motivating the paper: improving the allocation of scarce salesperson time in high-volume sales settings. To facilitate adoption and further research, we provide an open-source framework for building stopping agents at [redacted].

We next apply our proposed approach to real-world sales conversations and evaluate whether the resulting stopping agents can improve salesforce efficiency by reallocating time away from low-potential conversations and toward new sales opportunities.

4 Empirical Application

We study an outbound sales operation at a large telecommunications firm in Europe. The firm conducts cross-selling campaigns targeting its existing mobile subscribers. All sales calls were initiated through live phone calls by commissioned salespeople.⁶ The calls follow a standardized script designed to introduce and promote complementary products and services, though salespeople may deviate from the script when needed. Our analysis focuses on first-contact calls in which mobile subscribers are offered the opportunity to switch their electricity provider.

4.1 Data

The dataset covers a one-month period and includes 11,627 outbound sales calls placed by 79 different salespeople. For each call, we observe the complete anonymized conversation transcript, automatically transcribed and timestamped at the utterance level using a speech-to-text system based on the Whisper model [Radford et al., 2023]. We also observe the salesperson identifier, whether the call resulted in a sale based on the consumer confirming the energy contract (hereafter, call *success*), and call metadata including start and end times.

Table 1: Descriptive statistics.

	Min	Max	Mean	Standard Dev.
Call Success (Binary)	0	1	0.055	0.228
Call Duration (s)	60	3453	195	214
Failed Call Duration (s)	60	3453	169	163
Successful Call Duration (s)	79	3094	630	417
Salesperson Success Rate	0	16.9%	6.6%	4.4%
Salesperson Call Volume	1	830	147	186

Table 1 summarizes key descriptive statistics. Only 5.5% of calls succeeded. Failed calls accounted for 82% (517 hours) of the total call time and lasted 169 seconds on average. Successful calls were longer, averaging 630 seconds, reflecting the additional time required to complete the sale. The average salesperson made 147 calls and had a success rate of 6.6%. Thus, most salesperson time is spent on calls that ultimately fail, making this setting particularly well suited to studying whether low-potential conversations can be terminated earlier.

⁶Their compensation consists of base pay plus a bonus across a two-shift workday with a fixed number of hours. A large number of leads are assigned to each salesperson at random (i.e., salespeople do not choose their prospect pool).

For all subsequent analysis, we randomly partition the calls into a training set of 5,690 calls ($\sim 50\%$), a validation set of 3,499 calls ($\sim 30\%$), and a test (i.e., held-out) set of 2,438 calls ($\sim 20\%$). We stratify the split so that the call success rate ($\sim 5.5\%$) is similar across the three samples.

4.2 Training and Evaluating our Stopping Agent

We now describe how to train and evaluate our stopping agent using this dataset.

Reward configuration. We use the reward structure in Equation 1, which trades off the opportunity cost of salesperson time against the value of a completed sale. We set $b = 1$, so that each completed sale yields a unit reward, realized only if the agent never quits the call (i.e., $\tau = T$) and the call ends in a sale. We specify the time cost as $c(t) = \rho t$, where ρ represents the expected value of reallocating one unit of salesperson time to a new call. Under the assumption that new calls are drawn from the same distribution as the historical calls,⁷ we calibrate this opportunity cost using the training data as:

$$c(t) = \rho t = \frac{\text{Training success rate}}{\text{Average training call duration}} \times t. \quad (10)$$

Training and evaluation. Our main specification uses GPT-4.1 as the policy model within the estimation procedure developed in Section 3. Specifically, we use the OpenAI API to fine-tune the `gpt-4.1-2025-04-14` checkpoint on the recursively constructed prompt-response pairs illustrated in Figure 1, with training examples resampled according to their weights. We fine-tune for three epochs and select the epoch with the lowest validation loss. We then use the validation set to select the stopping threshold for each time period.

After training, we evaluate the resulting policy on the held-out test set by replaying each historical conversation and truncating it whenever the estimated policy chooses to stop. Since the policy is truncation-only, calls that are not stopped follow the historical conversation until the next time period or to completion.

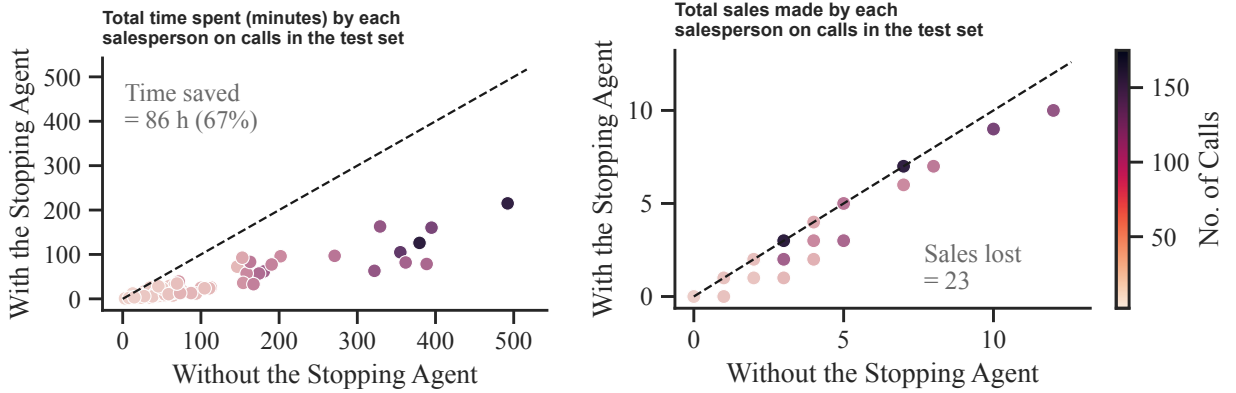
⁷This assumption is appropriate in high-volume sales settings with an ample supply of prospects. In other settings, the reward function can incorporate alternative benefits of ending low-potential calls earlier, such as lower labor costs, improved salesperson morale, or improved customer satisfaction.

4.3 Stopping Agent Performance

In this section, we evaluate stopping agents with opportunities to stop at $t \in \{60, 90\}$ and $t \in \{30, 60, 90\}$ seconds into the call.

To begin, we compare the *total time spent* and *total number of sales* by each salesperson with and without our $t \in \{30, 60, 90\}$ stopping agent (Figure 2). Each point represents a salesperson, with color intensity indicating their call volume in the test set. Points on the $x = y$ diagonal correspond to no change in time or sales from using the stopping agent. For instance, in the left panel, points below the diagonal are salespeople who would have saved time using our stopping agent.

Figure 2: Total time spent (left) and total number of sales made (right) by each salesperson, both with (y -axis) and without (x -axis) our stopping agent with stopping opportunities at $t \in \{30, 60, 90\}$.



Note: Each point corresponds to a salesperson. The $x = y$ diagonal corresponds to no time saved and no sales lost by using the stopping agent. Stopping agent outcomes are obtained by applying it to each salesperson’s test-set calls.

Salespeople spent 128 hours on these calls in total. Our stopping agent reduces total call time by 86 hours (67%) while retaining 109 of the 132 observed sales. At the salesperson level, time savings range from 5.3% to 93.0%.

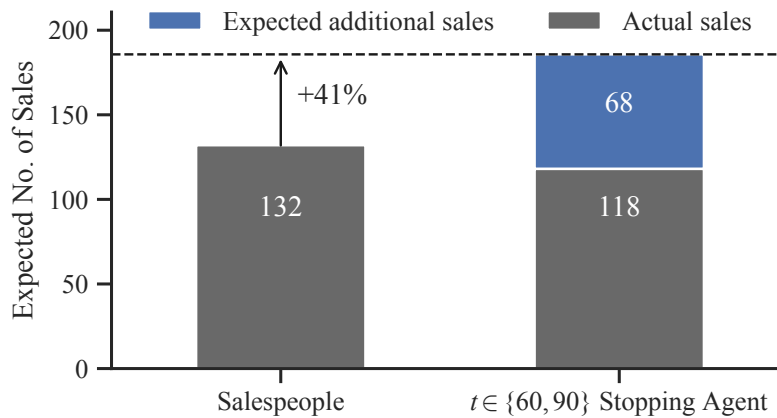
Stopping a call early trades off lost sales against salesperson time that can be reallocated to other prospects. Hence, we summarize the net effect using *expected sales*. We assume that time saved by stopping early is reallocated to new calls made by the same salesperson, drawn from the same distribution as that salesperson’s historical calls. Expected sales are therefore given by

$$\text{Expected sales} = \text{Actual sales} + \sum_s \frac{\text{Time saved}_s}{\text{Average call duration}_s} \times \text{Success rate}_s, \quad (11)$$

where the sum is over salespeople s . The quantities Average call duration _{s} and Success rate _{s} are calculated from salesperson s 's test-set calls, so the expected additional sales from reallocating saved time reflect each salesperson's own historical productivity. A salesperson with no sales in the test set therefore receives no expected additional sales from reallocating saved time.

We use this metric to first evaluate our stopping agent with stopping opportunities at $t \in \{60, 90\}$. This stopping agent retains 118 observed sales and saves 68 hours of salesperson time. Reallocating this saved time generates 68 additional expected sales, yielding a 41% increase in expected sales relative to the 132 sales observed without our stopping agent (Figure 3). Thus, even with only two relatively late stopping opportunities, our stopping agent can generate substantial gains in sales effectiveness.

Figure 3: Expected number of sales by our stopping agent with stopping opportunities at $t \in \{60, 90\}$.



Increasing the number of stopping opportunities. We next add a third stopping opportunity at $t = 30$ seconds and re-estimate the stopping policy. Allowing the agent to act earlier increases the total time saved from 68 to 86 hours, corresponding to a 67% reduction in call time. This policy retains 109 of the 132 observed sales, but the additional time available for new calls more than offsets these lost sales: expected sales increase by 47% relative to the salesperson baseline.

These gains are heterogeneous across the salesforce. As shown in Web Appendix D, our stopping agent increases expected sales by 150% for salespeople with success rates below the median, compared with 33% for those with above-median success rates. This finding suggests that algorithmic stopping support may be particularly valuable for less effective salespeople.

Figure 4: Expected number of sales by our stopping agent with stopping opportunities at $t \in \{30, 60, 90\}$.

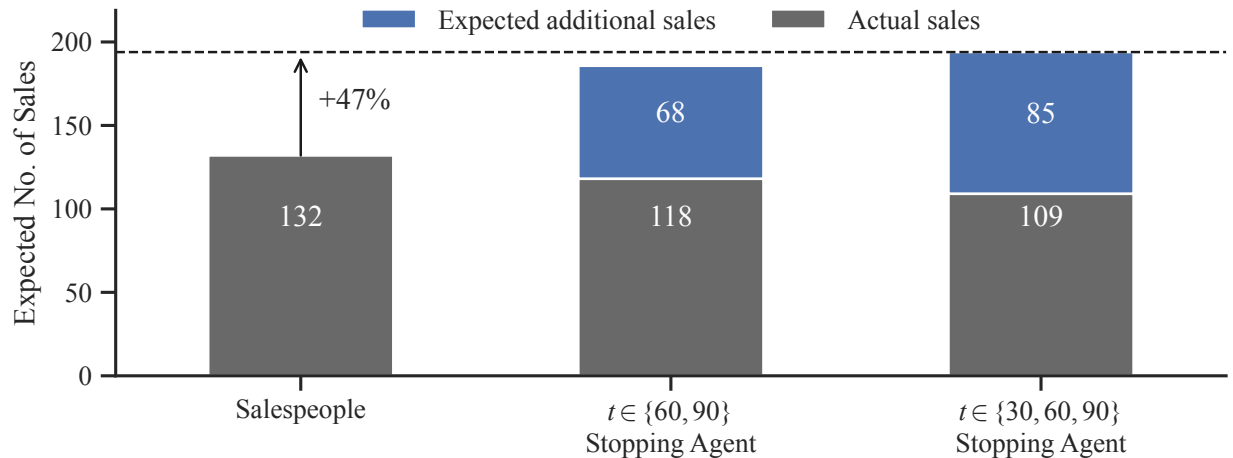


Figure 4 summarizes the comparison between salespeople and our stopping agents with stopping opportunities at $t \in \{60, 90\}$ and $t \in \{30, 60, 90\}$. Because the latter configuration generates the highest expected sales, we use $t \in \{30, 60, 90\}$ in the analyses that follow, although our methodology readily extends to more frequent stopping opportunities.⁸

We next explore how performance varies with the policy model used when estimating the stopping policy, and also evaluate alternative approaches to the stopping decision problem.

4.4 Understanding the Sources of Performance Gains

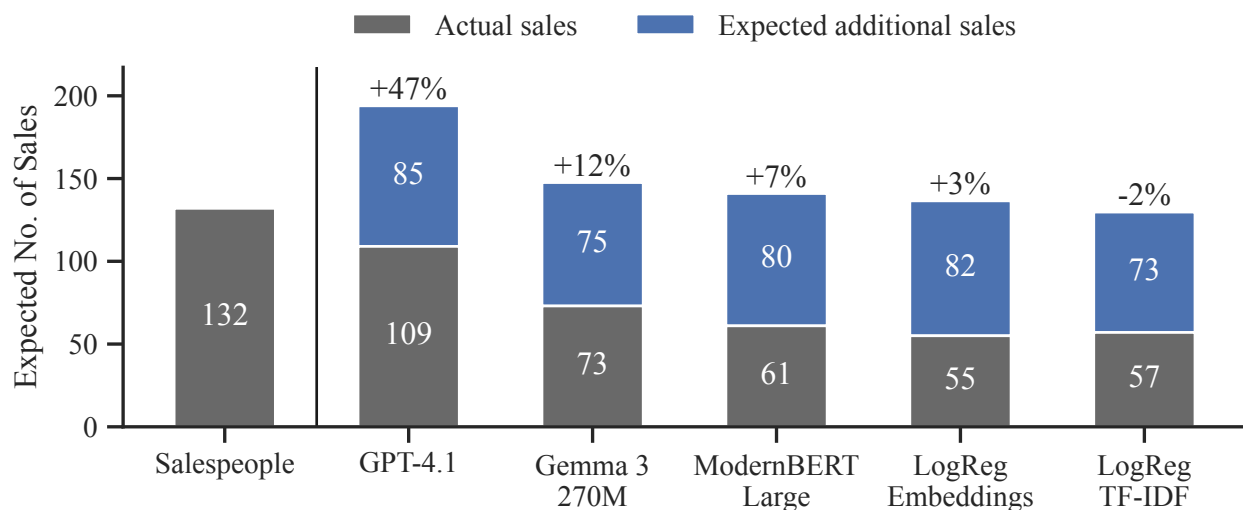
The performance of a stopping agent depends on both the policy model, which maps conversational text into stopping decisions, and the policy estimation procedure, which estimates the stopping policy from historical conversations. We first examine how performance varies with the policy model while holding the estimation procedure fixed. We then assess the value of formulating the problem as dynamic optimal stopping relative to simpler decision rules and compare our proposed policy estimation procedure with alternative optimal-stopping methods.

4.4.1 The Role of the Policy Model. A key determinant of a stopping agent’s performance is the policy model that maps the evolving conversation transcript into a stopping decision. We

⁸In practice, the number of stopping opportunities is limited by the granularity of the call transcription timestamps. Word-level timestamps offer the most granularity, because a quitting decision can be made after each word. However, real-time word-level transcription is more prone to errors and requires more computational resources.

isolate its contribution by estimating our stopping agent with a range of policy models while holding the policy estimation procedure fixed.

Figure 5: Stopping agents with alternative policy models.



Our results are presented in Figure 5. We start with the simplest policy model (rightmost bar): an L_2 -regularized logistic regression that maps the TF-IDF scores of the transcript’s unigrams and bigrams (i.e., a bag-of-words text representation) to a stopping score, with the regularization weight tuned via cross-validation. Using this policy model and text representation, our stopping agent *reduces* expected sales by 2%.

Holding the logistic regression policy model fixed and replacing the bag-of-words text representation with a 3072-dimensional OpenAI `text-embedding-3-large` embedding improves performance, yielding a 3% increase in expected sales. This comparison highlights the value of richer text representations: even a well-specified policy model may perform poorly when the conversational state is represented inadequately.

We next consider pretrained transformer models. We start with ModernBERT-large [Warner et al., 2024] — a bidirectional encoder-only transformer with 395 million parameters pretrained on 2 trillion tokens of English text and code. ModernBERT increases expected sales by 7% — roughly double the gain of the logistic regression-on-embeddings policy model. This suggests that pretrained transformers can improve stopping decisions beyond using fixed text representations.

Replacing ModernBERT with Gemma 3 270M [Gemma Team et al., 2025], a generative language model, raises the gain further, to 12%. While Gemma 3 270M is smaller than ModernBERT — 270 million parameters versus 395 million — it is pretrained on a larger corpus of 6 trillion tokens across more than 140 languages, including Spanish. The higher performance of Gemma 3 270M is consistent with the value of stronger and more relevant pretrained language representations. Finally, replacing Gemma 3 270M with GPT-4.1 increases expected sales by 47% — almost four times the gain obtained with Gemma 3 270M.

Overall, stopping performance improves systematically with the policy model’s natural language understanding capabilities. This pattern highlights the value of an estimation procedure that can accommodate increasingly capable LLMs, including proprietary API-based LLMs.⁹

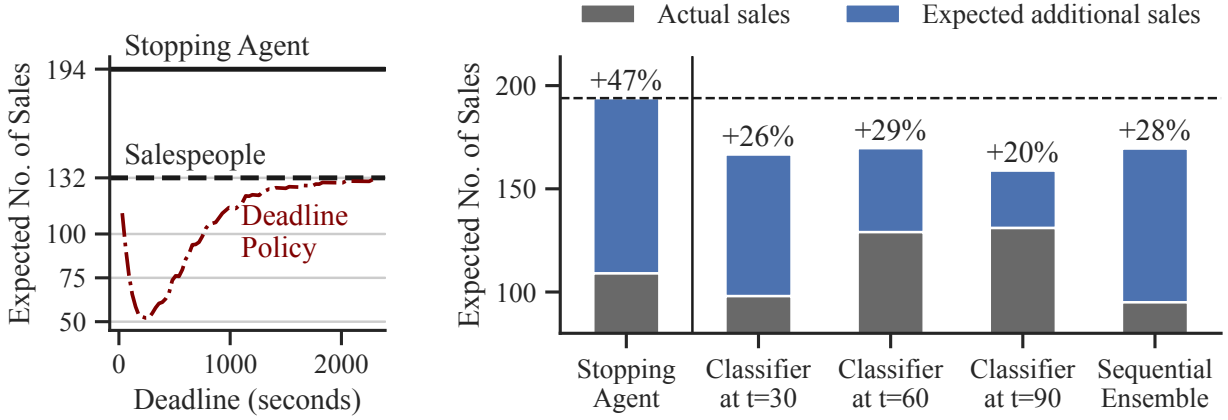
4.4.2 The Value of Dynamic Optimal Stopping. Stopping policies can alternatively be implemented using simpler decision rules that do not solve the full dynamic stopping problem. We compare our stopping agent with progressively more complex alternatives that either ignore the conversation content or make stopping decisions without explicitly optimizing future rewards.

A deadline policy. We begin with a *deadline policy* that quits at a fixed time t if the call has not yet succeeded by t . This truncation-only policy does not condition on the content of the conversation. Figure 6a presents expected sales across different deadlines, alongside the performance of both salespeople and our $t \in \{30, 60, 90\}$ stopping agent. Across all values of t , the deadline policy underperforms both salespeople and our stopping agent. Thus, the gains from our stopping agent do not arise simply from shortening calls; deciding which conversations to stop based on their content is important.

Classifier-based policies. Next, we evaluate a set of static classifier-based policies. Each policy consists of a GPT-4.1 LLM fine-tuned to *classify* whether the call will eventually fail to sell using

⁹Model capability by itself, however, may be insufficient without task-specific fine-tuning. Web Appendix C evaluates GPT-4.1, GPT-6 Luna, GPT-6 Sol, and GPT-6 Astra as zero-shot policy models (i.e., without fine-tuning) and finds that none matches the performance of our fine-tuned GPT-4.1 stopping agent.

Figure 6: Expected number of sales by our $t \in \{30, 60, 90\}$ stopping agent vs. several baselines.



(a) Expected no. of sales by our stopping agent and a deadline policy.

(b) Expected number of sales by our stopping agent, classifier-based policies at $t = 30, t = 60$, and $t = 90$, and a sequential classifier ensemble.

only the first $t = 30, t = 60$, or $t = 90$ seconds of the transcript, respectively.¹⁰ Each policy quits at t if the predicted failure probability exceeds a threshold, which we select to maximize the expected reward on the validation set. Unlike our stopping agent, these policies optimize a prediction problem—whether the call will eventually fail—and then translate that prediction into a single stopping decision at a fixed point in time. They therefore do not account for the option value of continuing the conversation and making another stopping decision later.

As Figure 6b shows, the classifier-based policies outperform salespeople but underperform our stopping agent despite using the same GPT-4.1 model. This comparison shows that strong outcome prediction alone is insufficient: explicitly solving the stopping problem yields better decisions than translating failure predictions into stopping rules.

A sequential classifier ensemble. Finally, we implement a stronger baseline that combines the classifier-based policies at $t = 30, t = 60$, and $t = 90$. The ensemble begins at $t = 30$: if the classifier-based policy at $t = 30$ does not quit, the decision is deferred to $t = 60$, and likewise to $t = 90$ if needed. This gives the ensemble the same sequence of stopping opportunities as our

¹⁰Specifically, we perform supervised fine-tuning on prompt-expected response pairs using the OpenAI API. As our prompt, we use the Spanish translation of “Here is the transcript of the first t seconds of a sales call between a consumer and a salesperson. <transcript> Will this call eventually end in a sale? (respond with ‘yes’ or ‘no’):”. Our expected response is ‘yes’ or ‘no’ based on whether the call actually ended in a sale. We train for 3 epochs and use the validation set for early stopping.

stopping agent, but each decision is still based on a prediction of eventual call failure rather than on the expected reward from stopping now relative to continuing. As Figure 6b shows, the sequential classifier ensemble also underperforms our stopping agent.

Overall, these comparisons show that neither shortening calls, predicting eventual failure, nor providing multiple stopping opportunities is sufficient. Our stopping agent performs better by explicitly optimizing the sequential stop-versus-continue decision.

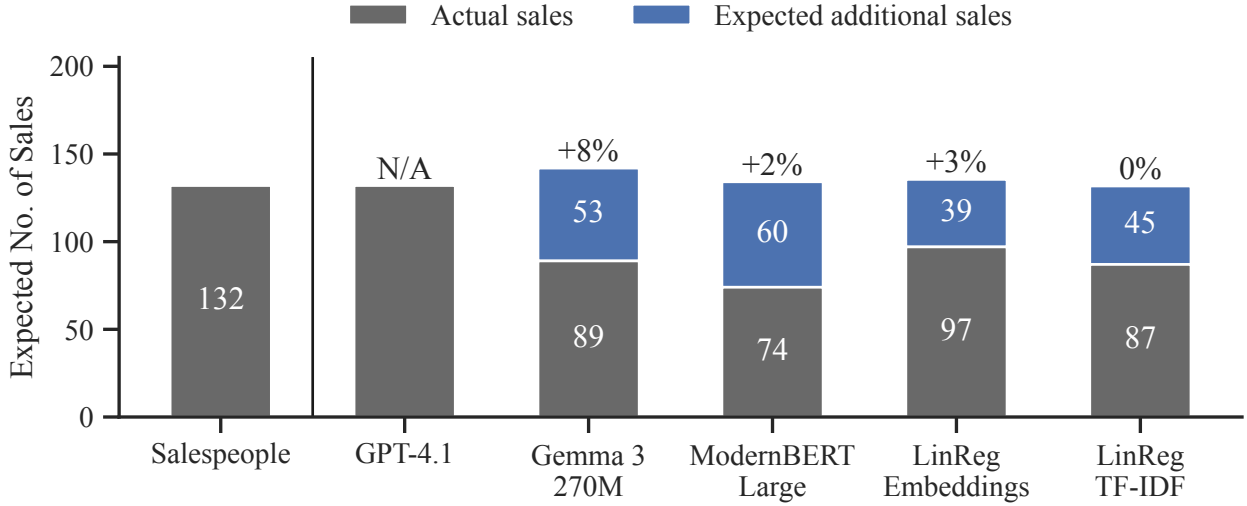
4.4.3 Comparison with Alternative Optimal-Stopping Methods. Finally, we compare our policy estimation procedure with alternative methods for learning optimal stopping policies from historical conversations.

Fitted Q -iteration. Our proposed procedure preserves the backward-induction logic of FQI but replaces numerical continuation-value regression with weighted cross-entropy loss minimization. To assess the empirical consequences of this reformulation, we implement FQI using a range of text representations and policy models paralleling those used with our proposed approach.

For the linear specifications, we replace the L_2 -regularized logistic regressions used in our approach with L_2 -regularized linear regressions. We consider both TF-IDF bag-of-words features and fixed text embeddings as representations of the conversational state. For the transformer-based specifications, we adapt ModernBERT and Gemma 3 270M to predict numerical continuation values: for ModernBERT, we replace the final classification layer with a regression layer, while for Gemma 3 270M, we replace the final language-model head with a regression head. In both cases, the transformer backbone is otherwise unchanged and all parameters are fine-tuned jointly by minimizing mean squared error. We do not implement FQI with GPT-4.1 because its fine-tuning API does not support the custom regression head and mean-squared-error objective required for continuation-value estimation. Figure 7 reports the results.

Overall, none of the FQI specifications approaches the performance of our GPT-4.1 stopping agent: the best FQI specification, using Gemma 3 270M, increases expected sales by 8%, compared with 47% under our proposed approach with GPT-4.1. Consistent with our earlier results, richer language models generally enable better stopping decisions, but the gains obtained under FQI remain substantially smaller than those achieved when the same transformer-based models are

Figure 7: Fitted Q -iteration (FQI) with alternative policy models.



trained using our proposed procedure. In particular, our approach increases expected sales from 2% to 7% with ModernBERT and from 8% to 12% with Gemma 3 270M, while also enabling the use of GPT-4.1, which cannot be implemented within FQI through OpenAI’s fine-tuning interface.

Deep optimal stopping methods. We also compare our policy estimation procedure with deep optimal stopping (DOS; Becker et al., 2019) and deep recurrent optimal stopping (DROS; Venkata and Bhattacharyya, 2023), two state-of-the-art approaches for estimating optimal stopping policies with neural networks. To facilitate comparison, we hold the text embeddings fixed across these methods and ours, and select each method’s specification using the validation set. The best DOS and DROS specifications increase expected sales by 2% and 4%, respectively. In comparison, our procedure achieves a 3% gain even with a simple logistic-regression policy applied to the same text embeddings, and substantially larger gains when paired with LLM policies. Web Appendix E provides implementation details and results across the full hyperparameter grid.

4.4.4 Summary. Taken together, these results indicate three sources of our stopping agent’s performance. First, more capable models of language improve our stopping agent’s ability to leverage decision-relevant information in conversational text. Second, explicitly formulating the problem as dynamic optimal stopping improves decisions relative to simpler rules based on

the elapsed call duration or predicted call outcomes. Third, the weighted cross-entropy loss is particularly valuable for transformer-based policy models, including LLMs.

Overall, our results suggest that our stopping agent’s gains arise from the combination of a strong language model, a dynamic decision objective, and an estimation procedure designed to connect the two.

4.5 Generalization and Real-Time Deployment Feasibility

We conclude our empirical evaluation by examining whether our stopping agent generalizes beyond the campaign on which it was estimated (without re-estimation on the new campaign), and whether its decisions can be generated quickly enough for real-time use.

4.5.1 Generalization to a Future Sales Campaign. We assess whether the estimated stopping policy generalizes beyond the marketing campaign on which it was estimated. Without re-estimating the stopping agent, we apply it to a separate campaign conducted six months later, involving a different subscriber population and largely different salespeople. This setting therefore provides a test of whether the estimated stopping policy transfers to a different operating environment without additional model estimation.

Our stopping agent continues to improve expected sales, yielding gains of 15.7% with stopping opportunities at $t \in \{60, 90\}$ and 16.4% with stopping opportunities at $t \in \{30, 60, 90\}$. Although these gains are smaller than in the original campaign, they remain economically meaningful even without re-estimation. Web Appendix F.1 describes the new campaign and reports the full results.

4.5.2 Decision Latency. Finally, we assess whether our stopping agent can generate decisions quickly enough for real-time deployment. Although the stopping agent is trained offline, it is intended to operate online as a conversation unfolds, evaluating the evolving transcript at each stopping opportunity. Real-time deployment therefore requires that inference be completed sufficiently quickly relative to the interval between successive decisions.

In our held-out test set, the median decision latency is 0.67 seconds with the GPT-4.1 policy model¹¹ and 26 milliseconds for the locally-hosted Gemma 3 270M policy model, both well below the 30-second interval between stopping opportunities. These results suggest that inference latency

¹¹This includes the round-trip network response time between our server and OpenAI’s.

is unlikely to be a binding constraint for real-time deployment in our setting. Web Appendix F.2 reports the full latency distributions.

5 Diagnosing Salespeople’s Quitting Decisions

Having demonstrated the performance of our stopping agent, we now examine patterns that may explain why salespeople’s quitting decisions fall short, using our stopping agent’s quitting decisions as a benchmark. Since identifying salespeople’s behavioral primitives directly is challenging without experimental data, we approach this question by empirically examining the behavioral patterns associated with salespeople’s quitting decisions.

Specifically, we examine two complementary questions. First, *what information do salespeople appear to use when deciding whether and when to quit?* We examine this question by training and analyzing interpretable machine learning models of a proxy for salespeople’s quitting decisions (Section 5.1). We find that salespeople appear to rely on simple decision rules that overweight a few salient phrases that often occur late in the call, whereas our stopping agent relies on a richer and dynamically changing set of linguistic cues. Second, *do salespeople adapt their quitting decisions when the opportunity cost of time changes?* In Section 5.2, we exploit variation in proximity to the end-of-shift over the workday as a proxy, and find that salespeople do not seem to sufficiently adjust their quitting decisions to account for call failure risk when time becomes more valuable.

Together, our results are consistent with salespeople facing cognitive constraints when attempting to disqualify consumers early in a call. One plausible explanation is that salespeople must simultaneously engage the consumer, respond to objections, and perform other selling tasks, which may limit their ability to integrate and act on multiple conversational signals in real time. These findings, while descriptive, highlight the potential value of algorithmic support for assisting with real-time qualification decisions.

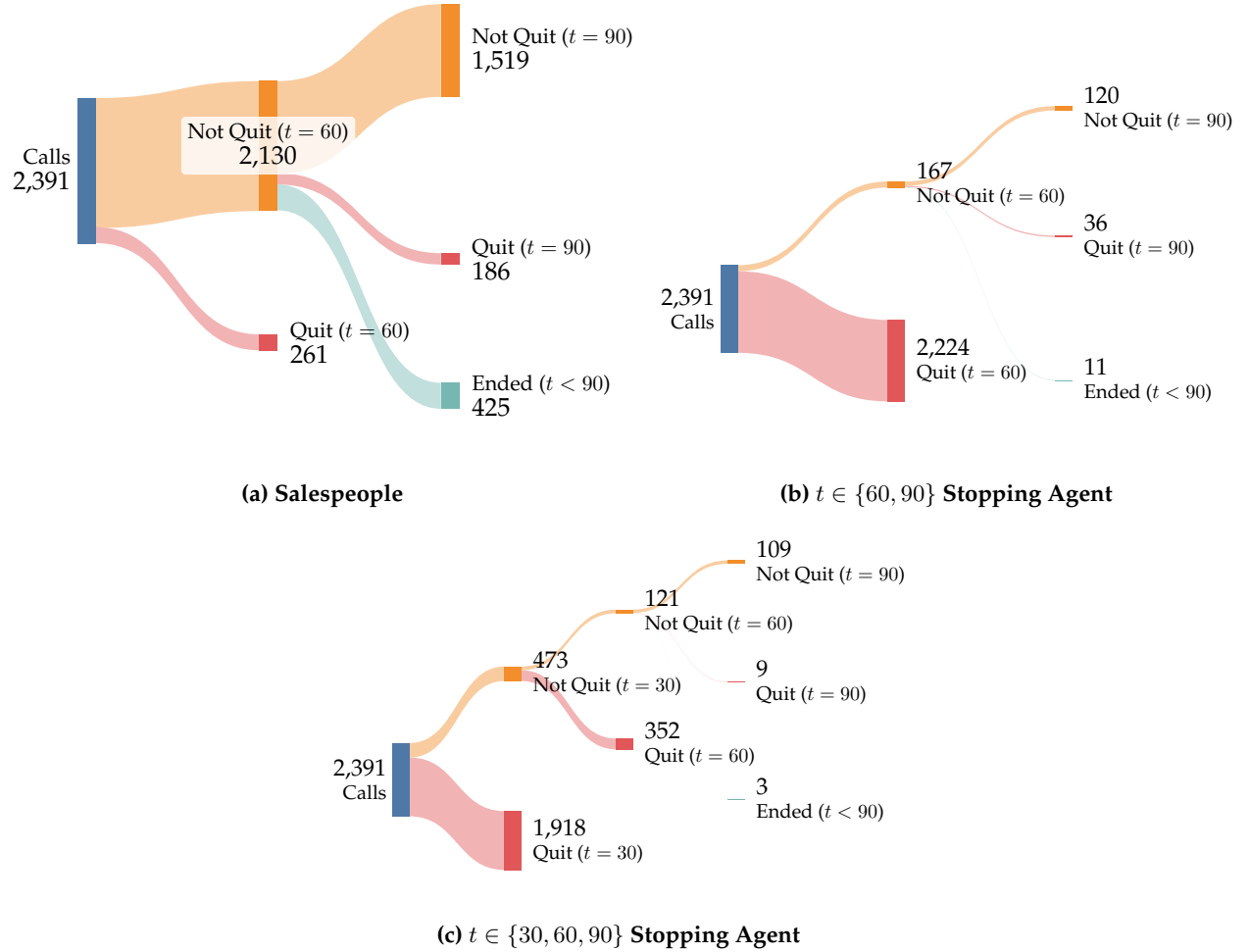
5.1 Explaining Quitting Decisions using Interpretable Machine Learning

5.1.1 Examining quitting times. To understand what information appears to drive salespeople’s quitting decisions, we begin by examining when salespeople and our stopping agents quit calls in the test set. To do so, we use a series of Sankey diagrams that depict quitting decisions

at discrete time points. To enable direct comparisons with our stopping agents, we encode the salesperson’s quitting decision at $t \in \{30, 60, 90\}$ as a binary indicator that equals 0 if the call lasts longer than $t + \Delta$ seconds, and 1 otherwise. We set $\Delta = 10$ to account for the average duration of closing salutations (though our results are robust to alternative values of Δ). If a call ends between $t + \Delta$ and the next time period, we label it as “ended” instead of “quit”.

Figure 8 visualizes quitting behavior using Sankey diagrams for (a) salespeople, (b) our $t \in \{60, 90\}$ stopping agent, and (c) our $t \in \{30, 60, 90\}$ stopping agent, respectively. Figure 8a shows

Figure 8: Sankey diagram of quitting decisions by salespeople and by our stopping agents.



Note: Salespeople did not quit any calls at $t = 30$, so we omit that decision opportunity from the Sankey diagram. 425 calls in the test set that were not quit by salespeople at $t = 60$ (by our definition) ended before $t = 90$.

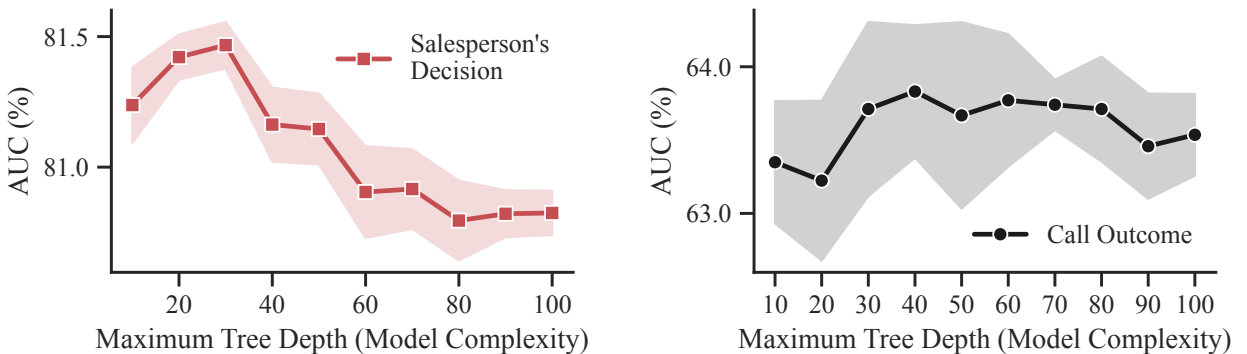
that salespeople hesitate to quit early: they quit *no* calls at $t = 30$, just 11% of calls at $t = 60$, and

just 11% of calls still ongoing at $t = 90$. In contrast, our $t \in \{60, 90\}$ stopping agent (Figure 8b) quits 93% of calls at $t = 60$, and 23% of calls still ongoing at $t = 90$. Our $t \in \{30, 60, 90\}$ stopping agent (Figure 8c) quits 80% of calls at $t = 30$, followed by 74% of calls still ongoing at $t = 60$ and 8% of calls still ongoing at $t = 90$. These patterns highlight a key behavioral difference between salespeople and our stopping agents: while our stopping agents favor quitting early, salespeople hesitate and delay.

We now turn to examining *why* salespeople hesitate to quit early. To answer this question, we empirically examine what linguistic signals appear to drive salespeople’s quitting decisions.

5.1.2 Examining the behavioral model underlying salespeople’s quitting decisions. The Sankey diagrams suggest that salespeople systematically delay quitting. We now ask what linguistic information appears to underlie these decisions. To better understand the behavioral model underlying salespeople’s quitting decisions, we train interpretable machine learning models to predict whether a salesperson will quit at $t = 60$ using only the call text before $t = 60$. Specifically, we train random forest classifiers with varying maximum depths (to control model complexity) on the training set calls. We use the 10,000 most frequent unigrams and bigrams as input features, and following standard practice, normalize the unigram and bigram term frequencies by their inverse document frequencies (i.e., TF-IDF).

Figure 9: Predicting salespeople quitting at $t = 60$ (left) and eventual call outcomes (right) given the first 60 seconds of the call transcript text using random forest models of different model complexities.



Note: Random forests with 1,000 trees are trained on unigrams and bigrams of the call transcript until $t = 60$. The reported AUC is averaged over 5 runs (shaded regions depict standard deviations).

Figure 9 (left) shows the held-out AUC of predicting salespeople’s quitting decisions with increasing model complexity. For comparison, Figure 9 (right) shows the held-out AUC of predicting eventual call outcomes given the same features and model. These figures reveal two key insights.

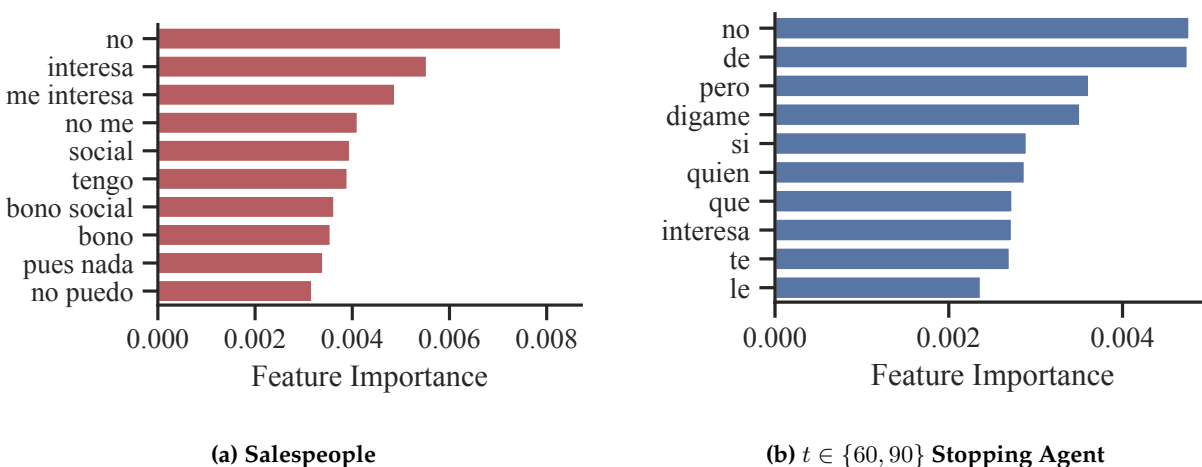
First, salespeople’s quitting decisions are substantially more predictable than call outcomes with random forests; the held-out AUC of predicting salespeople quitting at $t = 60$ is over 17 percentage points higher. Second, the performance of predicting salespeople’s quitting decisions deteriorates sharply with increasing model complexity (indicating overfitting), whereas the performance of predicting eventual call outcomes remains relatively stable with increasing model complexity.

These patterns suggest that salespeople’s quitting decisions follow simple rules, which can be captured by shallow random forests but are overfit by deeper ones. In contrast, call outcomes appear to follow more complex rules that are not easily captured even by deep random forests. Since the simple rules driving salespeople’s decisions may not reflect the complex rules that govern actual call outcomes, these findings suggest that salespeople are cognitively bounded. This may explain, in part, why salespeople’s quitting decisions underperform relative to our stopping agent.

5.1.3 Uncovering the linguistic cues potentially driving salespeople’s quitting decisions.

We now leverage the interpretability of random forests to uncover the linguistic cues that may drive salespeople’s quitting decisions. Specifically, we select the best random forest predictor of salespeople quitting at $t = 60$, and extract the most predictive features from this predictor based on their Gini importance [Breiman et al., 2017]. Figure 10a displays the 10 most predictive features.

Figure 10: Most important features (by Gini importance [Breiman et al., 2017]) in the best random forest model predicting quitting decisions at $t = 60$.



Notably, the most important features predicting salespeople quitting all come from the phrase “*no me interesa*” (i.e., “I’m not interested”), suggesting that salespeople tend to quit when they hear this explicit expression of disinterest, and tend to continue the call otherwise. Further, the feature importance distribution is skewed towards the features comprising “*no me interesa*”, indicating a disproportionate reliance on “*no me interesa*” over other phrases.

The phrase “I’m not interested” holds a special place in sales: it is the canonical example of an objection in sales training materials. Its prominence likely makes it *salient*, and salience is a well-documented source of bias in decision-making [Tversky and Kahneman, 1974]. However, “*no me interesa*” appears during the first 60 seconds of only 7.4% of the test set calls that ultimately fail. If salespeople over-rely on “*no me interesa*” to quit due to its salience, they may miss other less-salient indicators of failure that appear earlier in the call. Hence, waiting to hear “*no me interesa*” and ignoring other indicators may explain, in part, why salespeople delay quitting.

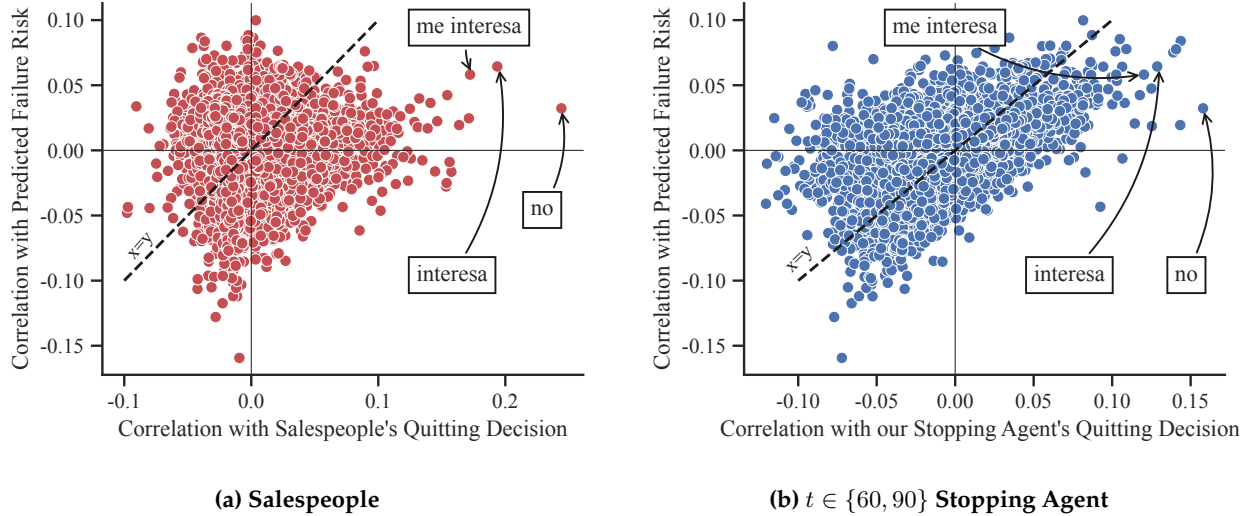
For comparison, we show the top 10 most predictive features from the best random forest predictor of our $t \in \{60, 90\}$ stopping agent’s quitting decisions at $t = 60$ in Figure 10b.¹² We note two observations. First, the distribution of feature importances is less skewed than in Figure 10a. Second, the most important features span several linguistic indicators of a lack of conversational progress. For example, “no”, “pero” (“but”), “digame” (“tell me”), and “quien” (“who”) indicate conversations stalled at the consumer identifying who the caller is and what they want, instead of progressing to affirmations of interest or to curiosity about the product. These phrases are more subtle indicators of disinterest than “*no me interesa*”, but appear early in calls that eventually failed.

Motivated by these results, we now further investigate whether salespeople systematically overweight “*no me interesa*” when deciding when to quit. We assess potential overweighting by comparing the correlation of the top 10,000 unigrams and bigrams with salespeople’s quitting decisions and with the predicted call failure risk at $t = 60$ measured using GPT-4.1 fine-tuned to predict eventual failure given the first 60 seconds of training call transcripts.

Figure 11a shows these correlations. The features comprising the phrase “*no me interesa*” are far more correlated with salespeople quitting ($r \in [0.17, 0.24]$) than with predicted call failure risk ($r \in [0.03, 0.06]$), and are clear outliers relative to the other unigrams and bigrams. This pattern

¹²Note that this random forest predictor is trained to predict our $t \in \{60, 90\}$ stopping agent’s quitting decisions at $t = 60$ on calls in the *validation set*, since our stopping agent was itself trained on calls in the training set.

Figure 11: Correlation of each of the top 10,000 unigrams and bigrams (red and blue points) with the call failure risk (predicted by fine-tuned GPT-4.1) at $t = 60$ (y -axis) and with quitting at $t = 60$ (x -axis) by (a) salespeople, and (b) our $t \in \{60, 90\}$ stopping agent.



suggests that “no me interesa” has an outsized influence on salespeople’s quitting decisions relative to other phrases and relative to its association with predicted call failure risk.

Panel (b) reports the analogous correlations for our $t \in \{60, 90\}$ stopping agent. In contrast to the salesperson panel, the features comprising “no me interesa” are no longer clear outliers: their associations with quitting are more similar to those of other phrases and lie relatively closer to the $y = x$ line. This pattern suggests that our stopping agent is less disproportionately influenced by this salient phrase and instead relies on a broader set of linguistic cues when deciding whether to quit at $t = 60$.

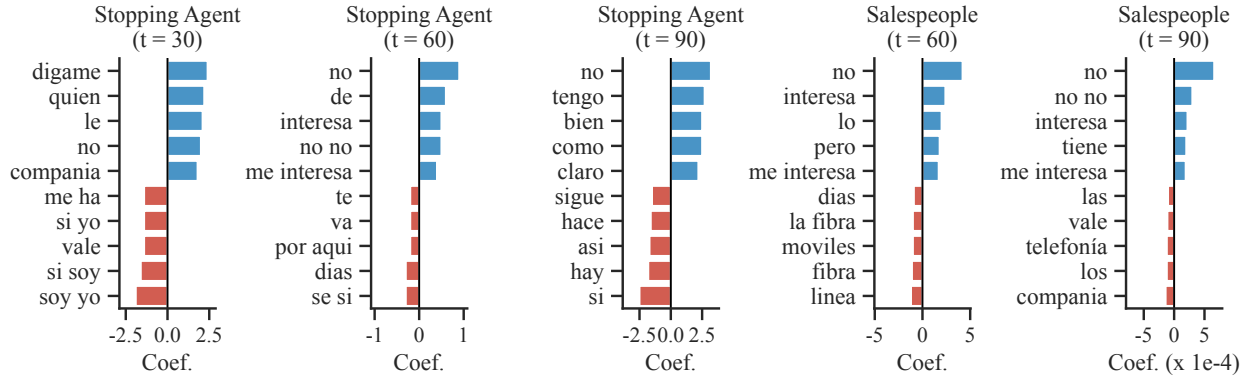
Examining dynamic variation in the linguistic cues associated with quitting. To examine how the linguistic cues associated with salespeople and our stopping agent quitting vary as the call progresses, we estimate penalized logistic regression models¹³ of our $t \in \{30, 60, 90\}$ stopping agent’s and the salespeople’s quitting decisions on calls in the test set at each $t \in \{30, 60, 90\}$ as a function of the top 10,000 unigrams and bigrams in the call transcript text before t .

In contrast with Figure 11, this analysis measures *conditional* correlations of each unigram and bigram with the quitting decision at t . Further, it measures the direction or sign of each association

¹³We employ L_2 penalization to alleviate potential overfitting due to the large number of correlated independent variables in the regression (i.e., 10,000 unigrams and bigrams).

(unlike the feature importances in Figure 10), thus complementing our previous analyses. We report the unigrams and bigrams most positively (top) and most negatively (bottom) associated with quitting at each t in Figure 12.

Figure 12: Top unigrams and bigrams (by logit coefficients) by decision-maker and decision opportunity.



Note: Coefficients are from a penalized logistic regression model of the stopping agent's or salesperson's quitting decision at $t \in \{30, 60, 90\}$ on the top 10,000 unigrams and bigrams in the test call transcript text before t . The top bars indicate the 5 unigrams and bigrams with the largest positive coefficients, and the bottom bars indicate the 5 unigrams and bigrams with the largest negative coefficients. Salespeople never quit at $t = 30$, so that decision opportunity is omitted. Where it appears, our partner firm's name is replaced with *compania* to maintain confidentiality.

Figure 12 reveals several insights. Consistent with Figure 10a, salespeople's quitting decisions at both $t = 60$ and $t = 90$ are overwhelmingly associated with the unigrams and bigrams comprising the phrase "*no me interesa*" ("I'm not interested"), suggesting that salespeople continue to wait for "*no me interesa*" across stopping opportunities. In contrast, the linguistic cues used by our stopping agent exhibit dynamic variation as the call progresses. Notably:

1. At $t = 30$, perfunctory responses such as "*digame*" ("tell me") and "*quien*" ("who?") are positively associated with our stopping agent quitting. The phrases "*soy yo*" ("it's me"), "*si soy*" ("yes, it's me"), and "*me ha*" ("it has"), however, signal affirmation and are negatively associated with our stopping agent quitting. These patterns suggest that, at $t = 30$, our stopping agent quits based on whether the salesperson is speaking to the right person.
2. At $t = 60$, the phrases positively associated with our stopping agent quitting shift to explicit indicators of disinterest, such as "*no*" and "*interesa*" ("interested"), while the negative associations now come from phrases that typically precede volunteering or asking for more information, such as "*se si*" ("if") and "*por aqui*" ("here"). Thus, between $t = 30$ and $t = 60$,

our stopping agent moves from quitting based on identifying *who* the salesperson is talking to, to assessing *what* the consumer’s degree of interest or disinterest is.

3. At $t = 90$, only 397 calls remain for our stopping agent to quit: the others are either quit by our stopping agent or by a salesperson before $t = 90$. The phrases positively associated with our stopping agent quitting at $t = 90$ include refusal (“no”), and constraint justifications such as “tengo” (“I have”), whereas the phrases most negatively associated with our stopping agent quitting all indicate conversational progress. Hence, at $t = 90$ our stopping agent quits based on *why* the prospect may not want the salesperson’s offering.

Overall, these results suggest that salespeople not only rely on a limited set of linguistic cues to decide whether and when to quit, but also do not update the cues they rely on as the conversation progresses. This is consistent with them being cognitively bounded. In contrast, our stopping agent uses a wide variety of linguistic cues, and exhibits dynamic variation in the linguistic cues it uses, to decide whether and when to quit.

5.2 Assessing Quitting Behavior Under Different Opportunity Costs of Time

The previous subsection characterized the information salespeople appear to use when deciding whether to quit. We now ask whether they adapt these decisions when the opportunity cost of time changes. Previous research shows that decision-making deficiencies are more pronounced during periods of high fatigue or pressure (e.g., Danziger et al. [2011] find that judges are less likely to grant parole near their meal break). Motivated by this research, we now analyze whether and how salespeople adjust their quitting behavior under higher time pressure or opportunity costs of time.

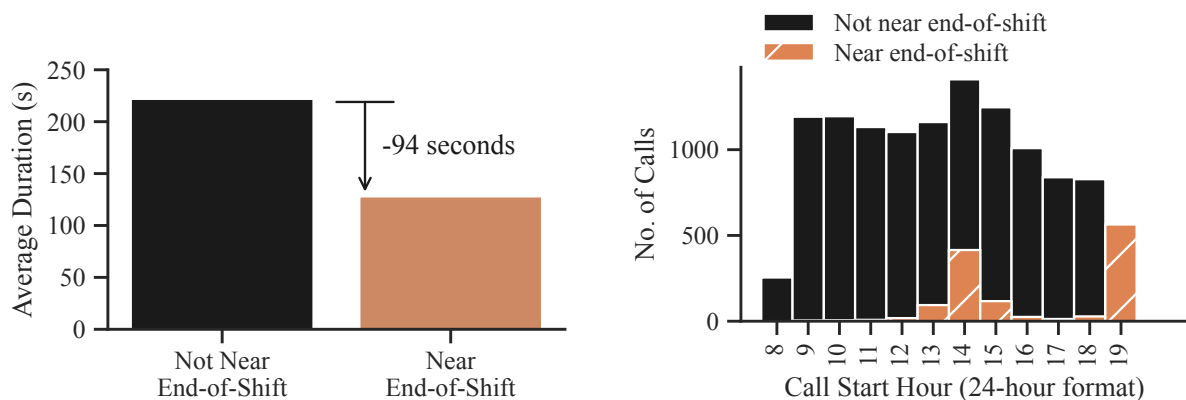
If salespeople accurately assess failure risk, higher opportunity costs of time should lead them to selectively shorten high-risk calls while preserving time for promising ones. If instead they shorten calls indiscriminately, this would suggest that they do not sufficiently incorporate failure risk into their quitting decisions.

Operationalizing variation in opportunity costs of time. To operationalize variation in opportunity costs of time, we use whether a call was made near the end of the salesperson’s shift as

a cost-shifter. Calls made near the end of the shift may have higher opportunity costs due to salespeople being eager to finish on time, avoid starting long conversations, or meet daily quotas.

We compute the time between the start of each call and the end of the salesperson’s shift, and bin test set calls into deciles based on their time-till-end-of-shift. We denote calls in the first decile as “near end-of-shift”. Figure 13a shows that “near end-of-shift” calls are 94 seconds shorter on average than earlier calls ($p < 0.001$), suggesting that salespeople indeed face higher opportunity costs of time during these calls. Figure 13b shows the calls’ start times and confirms that “near end-of-shift calls” are made in the last hour of salespeople’s shifts, which end at 3 p.m. and 8 p.m.

Figure 13: Whether a call was made near the end of the salesperson’s shift as a time cost-shifter.



(a) Avg. duration of “near end-of-shift” calls vs. others.

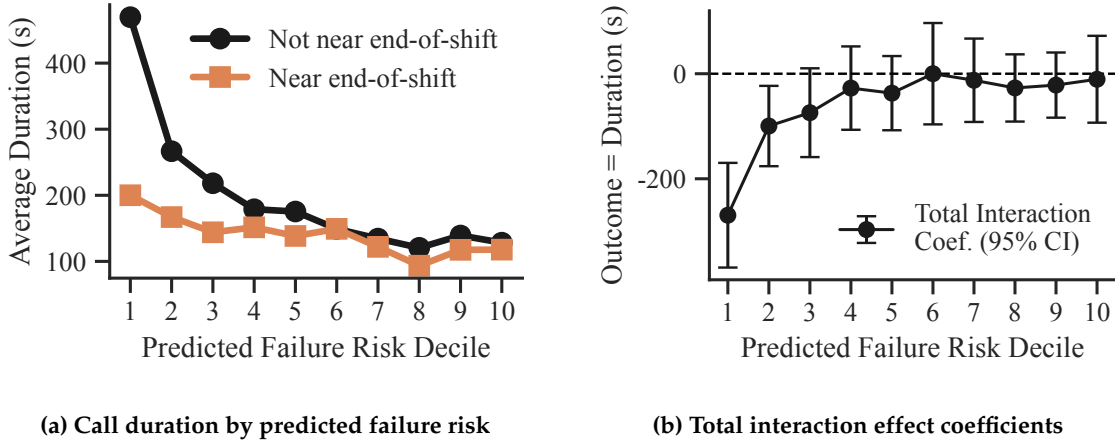
(b) Start times of “near end-of-shift” calls vs. others.

Note: The left plot shows the average duration of the “near end-of-shift” (i.e., in the first time-till-end-of-shift decile) calls and the other calls. The right plot shows the start time (hour) of the “near end-of-shift” calls and other calls.

Do salespeople allocate call time in a risk-sensitive manner? We now examine whether salespeople respond to higher opportunity costs of time by reallocating time more effectively across calls with different predicted failure risks measured using fine-tuned GPT-4.1. Figure 14a plots the average call duration by predicted failure risk decile (for test set calls), separately for near end-of-shift calls and for calls made earlier in the shift. If salespeople allocate time in a risk-sensitive manner, we would expect them to (1) spend less time on high-risk calls, and (2) differentially reduce the time spent on high-risk calls when the opportunity cost of time is higher.

Figure 14a shows that for calls made earlier in the shift, salespeople do spend less time on higher risk calls (i.e., the curve slopes downwards). However, near the end of their shift, salespeople shorten calls *throughout* the predicted failure risk distribution. In fact, they reduce time the most on calls in the lowest-risk deciles (1 and 2), calls that, *ex ante*, were more likely to succeed.

Figure 14: How salespeople allocate time across calls in the test set with varying predicted failure risks, comparing calls with a low (not near end of shift) vs. a high (near end of shift) opportunity cost of time.



Note: The predicted failure risk of a call is the probability of the call failing to end in a sale, binned into deciles, predicted by a fine-tuned GPT-4.1 model given the first 60 seconds of the call transcript. The right plot shows the total interaction effect coefficients and 95% confidence intervals from a regression of the call duration on the predicted failure risk decile interacted with a dummy for near end-of-shift calls (i.e., first time-till-end-of-shift decile).

To formally test this pattern, we estimate the following regression:

$$\text{Duration}_i = \alpha_0 + \gamma_0 s_i + \sum_{j=2}^{10} \alpha_j p_{ij} + \sum_{j=2}^{10} \beta_j p_{ij} s_i + \epsilon_i, \quad (12)$$

where p_{ij} is an indicator that call i falls in predicted failure risk decile j , and s_i is an indicator for the call being made near the end of the salesperson's shift. The coefficients β_j capture how the effect of being near the end of a shift varies across risk deciles.

Figure 14b shows the estimated total interaction coefficients (i.e., γ_0 for decile 1 and $\gamma_0 + \beta_j$ for deciles 2 to 10) and their 95% confidence intervals. We find statistically significant time reductions in low-failure-risk deciles (1 and 2) ($p < 0.001$), but no systematic shortening of calls in the other deciles. This pattern suggests that salespeople are not responding to higher time costs by

reducing the time spent on high-risk calls, but are instead shortening calls across the board, or even disproportionately shortening calls that were more promising *ex ante*.

5.3 Summary and Discussion

Our findings suggest that salespeople rely heavily on simple functions of a few salient cues (most notably, the phrase “no me interesa”) that often appear late in the call, and do not adjust the cues they rely on as the call progresses. Moreover, when the opportunity cost of time increases near the end of their shift, salespeople shorten calls broadly rather than concentrating those reductions among conversations with higher predicted failure risk.

Taken together, these patterns are consistent with cognitive constraints in integrating conversational information, failure risk, and the opportunity cost of time into real-time quitting decisions. One plausible explanation is that salespeople must make these judgments while simultaneously engaging consumers, responding to objections, and performing other selling tasks. The results therefore highlight the potential value of algorithmic decision support that can systematically process evolving conversational information and economic trade-offs when guiding qualification decisions.

6 Conclusion

In high-volume outbound sales settings, leads are ample yet most calls fail to end in a sale and consume substantial time and resources. This inefficiency poses a core managerial challenge: how to identify, in real time, which conversations are worth continuing and which should be abandoned to pursue other leads. We address this problem by introducing stopping agents: generative language agents trained via offline reinforcement learning to make sequential quit or wait decisions given the evolving transcript of each call to maximize the expected cumulative reward.

Our stopping agent delivers substantial gains. When applied to real-world sales conversations, our GPT-4.1 stopping agent reduces total call time by 67%. Reallocating the time saved to new calls increases expected sales by up to 47%. Our stopping agent yields sales gains even with a smaller open-weight policy model (Gemma 3 270M), a BERT-style policy model (ModernBERT), and a logistic regression policy model operating on text embeddings.

Notably, our stopping agent delivers these gains with a low decision latency using only the transcript text (without visuals or voice), and can be trained and deployed cost-effectively using either commercial APIs or open-weight LLMs. As such, our stopping agent offers a practical decision support tool that integrates with existing real-time call transcription workflows. We release open-source code that allows firms to implement our stopping agent on their own data.

Beyond performance improvements, our analysis offers insight into the behavioral foundations of salesperson inefficiency. We show that salespeople seem to rely on salient cues (e.g., “*no me interesa*”) that often appear late in the call, in contrast with our stopping agent that relies on subtle linguistic cues that appear earlier in the call, and that exhibits dynamic variation in the cues it relies on. These patterns suggest that cognitive bounds are a key constraint in dynamic qualification decisions, and that algorithmic decision support may help reduce the impact of these bounds.

Future work. This work opens several promising directions for future research. In this paper, we focus on short-term sales.¹⁴ Theoretically, one could easily adapt the reward structure to long-run objectives such as customer value or brand outcomes, which may encourage agents to persist longer in conversations towards achieving these objectives.

Another extension involves expanding the action space of our stopping agent. Given the versatility of the underlying language models, our approach could be extended to support a broader set of conversational decisions beyond quit and wait, for example, recommending scripted responses (à la Google’s smart-reply [Kannan et al., 2016]), asking clarifying questions, or suggesting pauses. Such multi-action policies would allow the agent to not only determine whether to quit, but also how to steer the conversation more effectively. However, this extension would require additional data from interactions between our stopping agent and salespeople (and the consumer’s responses), a more complex estimation procedure, and an off-policy evaluation protocol.

Our work focuses on the textual information of the call to guide the stopping agent’s decisions. Hence, our approach relies on high-quality transcripts and may be sensitive to inaccuracies in automatic speech recognition, transcription, and diarization. Foundation models for voice and speech are rapidly evolving (e.g., [Baevski et al., 2020]). Future research could leverage these models to directly incorporate voice and non-verbal signals (e.g., tone, pitch, and prosody) to

¹⁴Discussions with our partner firm indicate that short-term sales were indeed the goal of the campaigns we study.

enhance our stopping agents. Finally, understanding how salespeople respond to AI-generated recommendations remains an open managerial and behavioral question that could be addressed via field experimentation [Kawaguchi, 2021; Dietvorst et al., 2018].

Limitations. While our findings demonstrate the potential of stopping agents to improve sales-force productivity, several limitations warrant consideration. Our empirical setting involves one firm, one product category, and one language. While our stopping agent generalizes well within this context, its applicability to other industries, particularly those involving persuasive, consultative, or relationship-based selling, remains to be tested. Research on benchmarking and improving the persuasive ability of LLMs is nascent [Jin et al., 2024; Pauli et al., 2025], and how to generate persuasive language to optimize a long-term objective remains an open problem.

Our stopping agent is designed to save time while minimizing lost sales when salespeople spend too long on calls. Our empirical evidence from sales calls in the field demonstrates that spending too long on calls that eventually fail is indeed prevalent. However, our stopping agent is not designed to advise salespeople to persist in conversations that they decide to quit. Addressing this limitation is possible using offline-to-online reinforcement learning, where offline reinforcement learning is followed by online policy improvement through direct interactions with salespeople [Yue et al., 2024] in a field experiment. We leave this endeavour to future research.

Our stopping agent also inherits the limitations of the *reward hypothesis* underpinning all of reinforcement learning [Sutton et al., 1998], that “all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)”. Recent research has infused reinforcement learning algorithms with economic structure such as downward-sloping demand curves [Misra et al., 2019] and intertemporal budget constraints [Ko et al., 2024]. We demonstrate substantial gains without assuming such economic structure, and delegate exploration along this dimension to future work.

Summary. In sum, we show that the decision of whether and when to quit a sales conversation is not merely an intuitive judgment. Rather, it is a high-stakes optimization problem that can be solved effectively with AI. By surfacing systematic human errors and offering a scalable remedy,

stopping agents transform conversational data from a passive record to actionable input, paving the way for more intelligent, real-time decision support.

More broadly, our findings contribute to ongoing conversations in marketing and AI research. They illustrate how high-dimensional, unstructured data, such as live conversation transcripts, can be converted into actionable decision support through offline reinforcement learning. They also underscore the growing feasibility of embedding AI agents into frontline operational contexts, not as replacements for human workers, but as silent companions that improve outcomes in real time. Our approach offers a model for building behaviorally informed AI systems that enhance decision quality while revealing the cognitive limits of human judgment.

Declarations. *Funding and Competing Interests:* All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors acknowledge financial support from OpenAI.

References

- Alekh Agarwal, Nan Jiang, Sham M. Kakade, and Wen Sun. *Reinforcement Learning: Theory and Algorithms*. 2019. Online manuscript, <https://rltheorybook.github.io/>.
- Molly Ahearne, Mohsen Pourmasoudi, Yashar Atefi, and Son K Lam. Sales performance rankings: Examining the impact of the type of information displayed on sales force outcomes. *Journal of Marketing*, 89(1):94–116, 2025.
- Neeraj Arora, Ishita Chakraborty, and Yohei Nishimura. AI-human hybrids for marketing research: Leveraging large language models (LLMs) as collaborators. *Journal of Marketing*, 89(2):43–70, 2025.
- Eva Ascarza, Michael Ross, and Bruce GS Hardie. Why you aren’t getting more from your marketing AI. *Harvard Business Review*, 99(4):48–54, 2021.
- Alexei Baeovski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.
- Drew Bagnell, Byron Boots, and Sanjiban Choudhury. *Modern Adaptive Control and Reinforcement Learning*. 2023.
- Ebrahim Barzegary and Hema Yoganarasimhan. A recursive partitioning approach for dynamic discrete choice modeling in high dimensional settings. *arXiv preprint arXiv:2208.01476v2*, 2025.
- Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019.
- Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

- Raghu Bommaraju, S Arunachalam, and Sebastian Hohenberg. Multi-Segment and Single-Segment Sales Contests: Evidence of Their Effectiveness and the Underlying Mechanisms. *Journal of Marketing Research*, 62(3):447–465, 2025.
- Fernando Branco, Monic Sun, and J Miguel Villas-Boas. Too much information? Information provision and search costs. *Marketing Science*, 35(4):605–618, 2016.
- Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Chapman and Hall/CRC, 2017.
- Ishita Chakraborty, Khai Chiong, Howard Dover, and K Sudhir. Can AI and AI-Hybrids detect persuasion skills? Salesforce hiring with conversational video interviews. *Marketing Science*, 44(1):30–53, 2025.
- Jinglin Chen and Nan Jiang. Information-Theoretic Considerations in Batch Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1042–1051. PMLR, 2019.
- Doug J Chung, Thomas Steenburgh, and K Sudhir. Do bonuses enhance sales productivity? A dynamic structural analysis of bonus-based compensation plans. *Marketing Science*, 33(2):165–187, 2014.
- Øystein Daljord, Sanjog Misra, and Harikesh S Nair. Homogeneous contracts for heterogeneous agents: Aligning sales force composition and compensation. *Journal of Marketing Research*, 53(2):161–182, 2016.
- Shai Danziger, Jonathan Levav, and Liora Avnaim-Pesso. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, 108(17):6889–6892, 2011.
- Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 64(3):1155–1170, 2018.
- Matthew Dixon and Ted McKenna. Stop Losing Sales to Customer Indecision. *Harvard Business Review*, 2022.
- Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.
- Aishwarya Gemma Team, Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.
- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- Grand View. Call And Contact Center Outsourcing Market Size, Share & Trends Analysis Report By Type (Voice, Chat Support), By Outsourcing Type, By Services, By Enterprise Size, By End-use, By Region, And Segment Forecasts, 2025 - 2030. <https://www.grandviewresearch.com/industry-analysis/call-contact-center-outsourcing-market-report>, 2024.

- Liang Guo. Strategic communication before price haggling: A tale of two orientations. *Marketing Science*, 41(5):922–940, 2022.
- Johannes Habel, Michael Ahearne, Seshadri Tirunillai, and Amy Vandaveer Novak. Do AI Role Plays Improve Sales Performance? Evidence from a Longitudinal Field Study. *Available at SSRN 5717822*, 2025.
- Saiquan Hu, Juanjuan Zhang, and Yuting Zhu. Beyond Zero: Jump-Starting Sales With a Recommender System for Missing-By-Choice Data. *MIT Sloan Research Paper*, 2024.
- Sam K Hui, Jehoshua Eliashberg, and Edward I George. Modeling DVD preorder and sales: An optimal stopping approach. *Marketing Science*, 27(6):1097–1110, 2008.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.
- Chuhao Jin, Kening Ren, Lingzhen Kong, Xiting Wang, Ruihua Song, and Huan Chen. Persuading across Diverse Domains: a Dataset and Persuasion Large Language Model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1678–1706. 2024.
- Enoch Hyunwook Kang, Hema Yoganarasimhan, and Lalit Jain. An Empirical Risk Minimization Approach for Offline Inverse Reinforcement Learning and Dynamic Discrete Choice Models. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pages 341–341. 2025.
- Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, et al. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 955–964. 2016.
- Yael Karlinsky-Shichor and Oded Netzer. Automating the B2B salesperson pricing decisions: A human-machine hybrid approach. *Marketing Science*, 43(1):138–157, 2024.
- Kohei Kawaguchi. When will workers follow an algorithm? A field experiment with a retail business. *Management Science*, 67(3):1670–1695, 2021.
- Minkyung Kim, K Sudhir, and Kosuke Uetake. A structural model of a multitasking salesforce: Incentives, private information, and job design. *Management Science*, 68(6):4602–4630, 2022.
- Minkyung Kim, K Sudhir, Kosuke Uetake, and Rodrigo Canales. When salespeople manage customer relationships: Multidimensional incentives and private information. *Journal of Marketing Research*, 56(5):749–766, 2019.
- Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. 2015.
- Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.
- Ryuya Ko, Kosuke Uetake, Kohei Yata, and Ryosuke Okada. When to target customers? retention management using dynamic off-policy policy learning. *Available at SSRN*, 2024.

- Nathan Lambert. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*, 2025.
- Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: A simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.
- Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. Better autoregressive regression with LLMs via regression-aware fine-tuning. In *International Conference on Learning Representations*, volume 2025, pages 69871–69901. 2025.
- Kanishka Misra, Eric M Schwartz, and Jacob Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252, 2019.
- Sanjog Misra. Selling and sales management. In *Handbook of the Economics of Marketing*, volume 1, pages 441–496. Elsevier, 2019.
- Sanjog Misra and Harikesh S Nair. A structural model of sales-force compensation dynamics: Estimation and field implementation. *Quantitative Marketing and Economics*, 9:211–257, 2011.
- Sendhil Mullainathan and Ziad Obermeyer. Diagnosing physician error: A machine learning approach to low-value health care. *The Quarterly Journal of Economics*, 137(2):679–727, 2022.
- Amalie Brogaard Pauli, Isabelle Augenstein, and Ira Assent. Measuring and Benchmarking Large Language Models’ Capabilities to Generate Persuasive Language. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10056–10075. Association for Computational Linguistics, Albuquerque, New Mexico, 2025. ISBN 979-8-89176-189-6.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.
- Ashesh Rambachan. Identifying prediction mistakes in observational data. *The Quarterly Journal of Economics*, 139(3):1665–1711, 2024.
- James C Reeder III, Nawar Chaker, and Johannes Habel. Improving Sales Forecasting: Leveraging Unstructured CRM Activity Logs, Large Language Models, and Generative AI. *Available at SSRN*, 2024.
- John Rust. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica: Journal of the Econometric Society*, pages 999–1033, 1987.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Albert N. Shiryaev. *Optimal Stopping Rules*, volume 8 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, Heidelberg, 1978. ISBN 978-3-540-74010-0.

- Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.
- Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- John N Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.
- Amos Tversky and Daniel Kahneman. Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *Science*, 185(4157):1124–1131, 1974.
- Niranjan Damera Venkata and Chiranjib Bhattacharyya. Deep recurrent optimal stopping. *Advances in Neural Information Processing Systems*, 36:12222–12244, 2023.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. 2024.
- Hema Yoganarasimhan. The value of reputation in an online freelance marketplace. *Marketing Science*, 32(6):860–891, 2013.
- Sheng Yue, Xingyuan Hua, Ju Ren, Sen Lin, Junshan Zhang, and Yaoxue Zhang. OLLIE: Imitation Learning from Offline Pretraining to Online Finetuning. In *International Conference on Machine Learning*, pages 57966–58018. PMLR, 2024.
- Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.
- Rami Zwick, Amnon Rapoport, Alison King Chung Lo, and AV Muthukrishnan. Consumer sequential search: Not enough or too much? *Marketing Science*, 22(4):503–519, 2003.

A Theoretical Justification

We now show that, at the population level and assuming a sufficiently expressive policy class¹⁵, the population analogue of Algorithm 2 approximates the maximum expected cumulative reward in Equation (2). Our proof strategy mirrors Becker et al. [2019], and our argument proceeds as follows.

First, holding the already-estimated downstream stopping policies fixed, we characterize the current-period stopping decision that maximizes the expected reward. Second, we show that thresholding the exact minimizer of our weighted cross-entropy loss yields this expected-reward-maximizing decision. Finally, we apply this argument recursively backwards in time to establish that the complete stopping policy comes within ε of the maximum expected cumulative reward for any $\varepsilon > 0$. This result bounds the suboptimality of the algorithm.

Throughout the appendix, expectations are taken over the population distribution from which the historical conversations are drawn. We assume that rewards are integrable and that stopping policies are measurable functions of the conversational state. These population results do not address finite-sample estimation error, optimization error, model misspecification, or the validation-based threshold selection used in the empirical application.

A.1 Characterizing the Optimal Current-Period Stopping Decision

Fix a time $t \in \{1, \dots, T-1\}$ and suppose that the downstream stopping policies $\pi_{t+1}^{\theta_{t+1}}, \dots, \pi_T^{\theta_T}$ have already been estimated, with $\pi_T^{\theta_T}(s) = 1$ for every terminal state. These policies induce the downstream stopping time $\tau_{t+1} \in \{t+1, \dots, T\}$, defined as the first time after t at which the downstream policy stops the conversation.

If the current-period policy $\pi_t^{\theta_t}$ stops at t , it receives $g(t, s_t)$. If it continues, it subsequently stops at τ_{t+1} and receives $g(\tau_{t+1}, s_{\tau_{t+1}})$. Define the stopping advantage:

$$\Delta_t = g(t, s_t) - g(\tau_{t+1}, s_{\tau_{t+1}}),$$

which is the population counterpart of Δ_t^k in Algorithm 2. Using the stopping advantage, the expected reward from time t onward under any current stopping policy $\pi_t^{\theta_t}$ is:

$$\begin{aligned} & \mathbb{E} \left[g(t, s_t) \pi_t^{\theta_t}(s_t) + g(\tau_{t+1}, s_{\tau_{t+1}}) (1 - \pi_t^{\theta_t}(s_t)) \right] \\ &= \mathbb{E} \left[g(\tau_{t+1}, s_{\tau_{t+1}}) + \Delta_t \pi_t^{\theta_t}(s_t) \right]. \end{aligned} \tag{13}$$

Conditional on the current state $s_t = s$, the current decision affects expected reward only through $\pi_t^{\theta_t}(s) \mathbb{E}[\Delta_t \mid s_t = s]$. Because $\pi_t^{\theta_t}(s) \in \{0, 1\}$, it maximizes expected reward by stopping when $\mathbb{E}[\Delta_t \mid s_t = s]$ is positive and continuing otherwise. Holding the downstream stopping

¹⁵To control function approximation error, Becker et al. [2019] rely on the universal approximation properties of neural networks, whereas we rely on the empirically demonstrated ability of LLMs to represent natural language.

policies fixed, an expected-reward-maximizing current-period policy is given by:

$$\pi_t^*(s) = \mathbb{1}[\mathbb{E}[\Delta_t \mid s_t = s] > 0]. \quad (14)$$

A.2 Weighted Cross-Entropy Loss Minimization Recovers the Optimal Current-Period Stopping Decision

Let $y_t = \mathbb{1}[\Delta_t > 0]$ and $w_t = |\Delta_t|$ denote the corresponding target and weight in Algorithm 2. Conditional on $s_t = s$, the conditional weighted cross-entropy loss is:

$$-\mathbb{E} [w_t \{y_t \log F_t(s) + (1 - y_t) \log(1 - F_t(s))\} \mid s_t = s]. \quad (15)$$

where $F_t(s) \in [0, 1]$ is the stopping score for state s at time t .

Because $F_t(s)$ is fixed conditional on $s_t = s$, Equation (15) can first be written as:

$$-\mathbb{E}[w_t y_t \mid s_t = s] \log F_t(s) - \mathbb{E}[w_t (1 - y_t) \mid s_t = s] \log(1 - F_t(s)).$$

Minimizing this expression with respect to $F_t(s)$ gives the ideal stopping score:

$$F_t^*(s) = \frac{\mathbb{E}[w_t y_t \mid s_t = s]}{\mathbb{E}[w_t \mid s_t = s]} = \frac{\mathbb{E}[\Delta_t \mathbb{1}[\Delta_t > 0] \mid s_t = s]}{\mathbb{E}[|\Delta_t| \mid s_t = s]}. \quad (16)$$

We threshold $F_t^*(s)$ at $1/2^\dagger$ to obtain the stopping decision $\pi_t(s) = \mathbb{1}[F_t^*(s) > 1/2]$, which implies:

$$\begin{aligned} F_t^*(s) > \frac{1}{2} &\iff 2\mathbb{E}[\Delta_t \mathbb{1}[\Delta_t > 0] \mid s_t = s] > \mathbb{E}[|\Delta_t| \mid s_t = s] \\ &\iff \mathbb{E}[\Delta_t \mathbb{1}[\Delta_t > 0] \mid s_t = s] > \mathbb{E}[-\Delta_t \mathbb{1}[\Delta_t < 0] \mid s_t = s] \\ &\iff \mathbb{E}[\Delta_t \mid s_t = s] > 0. \end{aligned} \quad (17)$$

Thus, conditional on the fixed downstream policy, the stopping decision obtained from the exact minimizer of the weighted cross-entropy loss is identical to the expected-reward-maximizing policy in Equation (14).

In practice, an estimated policy may fail to implement this population best-response decision exactly. The next subsection shows how the resulting per-period expected-reward approximation errors propagate through the backward recursion.

A.3 Propagation of Per-Period Expected-Reward Approximation Errors

Let V_t^* denote the maximum expected reward achievable from time t onward:

$$V_t^* = \sup_{t \leq \tau \leq T} \mathbb{E}[g(\tau, s_\tau)]. \quad (18)$$

[†]In practice, we pick a threshold κ_t at each t using a validation set due to its better empirical performance.

Fix $t < T$ and let τ_{t+1} be the stopping time induced by the already-estimated downstream policy, which may be suboptimal. Its suboptimality is given by:

$$\varepsilon_{t+1} = V_{t+1}^* - \mathbb{E}[g(\tau_{t+1}, s_{\tau_{t+1}})] . \quad (19)$$

Appendix A.1 characterizes the expected-reward-maximizing current-period decision, and Appendix A.2 shows that thresholding the exact minimizer of our weighted cross-entropy loss yields this decision. Let π_t be a current-period policy whose expected reward is lower than that under the ideal current-period decision, holding the downstream policy fixed, by at most δ_t . Together with the downstream policy, π_t forms the extended stopping policy, which induces the stopping time:

$$\tau_t = t\pi_t(s_t) + \tau_{t+1}(1 - \pi_t(s_t)) . \quad (20)$$

Let τ_t^* denote the stopping time induced by the ideal current-period decision and the already-estimated downstream policy (with suboptimality ε_{t+1}). Theorem 1 of Becker et al. [2019] gives:

$$\mathbb{E}[g(\tau_t^*, s_{\tau_t^*})] \geq V_t^* - \underbrace{(V_{t+1}^* - \mathbb{E}[g(\tau_{t+1}, s_{\tau_{t+1}})])}_{\varepsilon_{t+1}} . \quad (21a)$$

By the definition of δ_t :

$$\mathbb{E}[g(\tau_t^*, s_{\tau_t^*})] - \mathbb{E}[g(\tau_t, s_{\tau_t})] \leq \delta_t \implies \mathbb{E}[g(\tau_t, s_{\tau_t})] \geq \mathbb{E}[g(\tau_t^*, s_{\tau_t^*})] - \delta_t . \quad (21b)$$

Combining these two inequalities gives:

$$\mathbb{E}[g(\tau_t, s_{\tau_t})] \geq V_t^* - \varepsilon_{t+1} - \delta_t . \quad (21c)$$

Equivalently, defining $\varepsilon_t = V_t^* - \mathbb{E}[g(\tau_t, s_{\tau_t})]$, Equation (21c) implies $\varepsilon_t \leq \varepsilon_{t+1} + \delta_t$. Thus, extending the policy backwards by one period can increase its suboptimality by at most δ_t . Since the policy class is sufficiently expressive, each per-period reward gap δ_t can be made arbitrarily small.

At the terminal time, $\tau_T = T$ and $\varepsilon_T = 0$. For any $\varepsilon > 0$, we can therefore make the current-period reward gaps small enough that $\sum_{t=1}^{T-1} \delta_t \leq \varepsilon$. Applying Equation (21c) recursively for $t = T - 1, \dots, 1$ then gives $\varepsilon_1 \leq \varepsilon$. Hence, the population analogue of Algorithm 2 can attain expected reward within ε of the maximum in Equation (2).

Relationship to the empirical procedure. Although our theoretical analysis considers the population setting, it provides a theoretical justification for Algorithm 2. In particular, it establishes that the weighted cross-entropy objective is aligned with the expected-reward-maximizing current-period decision, and that Algorithm 2 approximates the expected cumulative reward in Equation (2). Algorithm 2 operationalizes this population-level justification in finite samples with LLM policies,

translating the theoretical characterization into a practical estimation strategy. We delegate the development of finite-sample guarantees for LLM-parameterized stopping policies to future work.

B Vanishing Gradients in Deep Optimal Stopping (DOS)

B.1 Theoretical Derivation of Vanishing Gradients

Like our proposed approach and FQI, DOS [Becker et al., 2019] proceeds backwards in time from the terminal period. At each time t , DOS treats the stopping policies already estimated for $t + 1, \dots, T$ as fixed and uses them to determine the downstream stopping time for each trajectory. It then compares the reward from stopping at t with the reward from continuing and following the previously-estimated downstream policy before estimating the stopping policy for time t .

DOS estimates each θ_t , $t < T$, by minimizing the following [Becker et al., 2019, Equation 15]:

$$\hat{\theta}_t^{\text{DOS}} = \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M \left[g(t, s_t^k) F_t^{\theta_t}(s_t^k) + g(\tau_{t+1}^k, s_{\tau_{t+1}}^k) (1 - F_t^{\theta_t}(s_t^k)) \right]. \quad (22)$$

We rewrite Equation (22) using the stopping advantage $\Delta_t^k = g(t, s_t^k) - g(\tau_{t+1}^k, s_{\tau_{t+1}}^k)$:

$$\hat{\theta}_t^{\text{DOS}} = \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M g(\tau_{t+1}^k, s_{\tau_{t+1}}^k) - \frac{1}{M} \sum_{k=1}^M \Delta_t^k F_t^{\theta_t}(s_t^k). \quad (23)$$

Since the first term in Equation (23) is a constant, the minimizer does not depend on it:

$$\begin{aligned} \hat{\theta}_t^{\text{DOS}} &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M \Delta_t^k F_t^{\theta_t}(s_t^k) \\ &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k (2y_t^k - 1) F_t^{\theta_t}(s_t^k) \\ &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k (2y_t^k - 1) F_t^{\theta_t}(s_t^k) - \frac{1}{M} \sum_{k=1}^M w_t^k (1 - y_t^k) \\ &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k \left[(2y_t^k - 1) F_t^{\theta_t}(s_t^k) + (1 - y_t^k) \right] \\ &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k \left[2y_t^k F_t^{\theta_t}(s_t^k) - F_t^{\theta_t}(s_t^k) + 1 - y_t^k \right] \\ &= \arg \min_{\theta_t} -\frac{1}{M} \sum_{k=1}^M w_t^k \left[y_t^k F_t^{\theta_t}(s_t^k) + (1 - y_t^k) (1 - F_t^{\theta_t}(s_t^k)) \right], \end{aligned} \quad (24)$$

where the third line adds $-\frac{1}{M} \sum_{k=1}^M w_t^k (1 - y_t^k)$, which does not change the minimizer.

Equation (24) reveals that deep optimal stopping is equivalent to minimizing the weighted mean absolute error (MAE) loss for the targets y_t^k , since

$$y_t^k F_t^{\theta_t}(s_t^k) + (1 - y_t^k)(1 - F_t^{\theta_t}(s_t^k)) = 1 - |y_t^k - F_t^{\theta_t}(s_t^k)|.$$

The MAE loss presents optimization difficulties in practice due to vanishing gradients [Zhang and Sabuncu, 2018, Equation 5]. Taking the gradient of the objective in Equation (24) gives:

$$\nabla_{\theta_t} \left\{ -\frac{1}{M} \sum_{k=1}^M w_t^k \left[y_t^k F_t^{\theta_t}(s_t^k) + (1 - y_t^k)(1 - F_t^{\theta_t}(s_t^k)) \right] \right\} = -\frac{1}{M} \sum_{k=1}^M w_t^k (2y_t^k - 1) \nabla_{\theta_t} F_t^{\theta_t}(s_t^k). \quad (25)$$

Since DOS uses a sigmoid output layer to scale $F_t^{\theta_t}$ between 0 and 1, $\nabla_{\theta_t} F_t^{\theta_t}(s_t^k)$ contains the multiplicative factor

$$F_t^{\theta_t}(s_t^k)(1 - F_t^{\theta_t}(s_t^k)).$$

This multiplicative factor approaches 0 as $F_t^{\theta_t}(s_t^k)$ approaches 0 or 1. Hence, the gradient vanishes whenever $F_t^{\theta_t}(s_t^k)$ is highly confident, including when it is confidently incorrect (e.g., $F_t^{\theta_t}(s_t^k) \rightarrow 1$ when $y_t^k = 0$).

B.2 Empirical Demonstration of Vanishing Gradients

We now compare the finite-sample performance of our proposed approach with DOS using the Bermudan max-call option pricing simulation environment from Becker et al. [2019]. We implement DOS exactly as proposed in Becker et al. [2019] and compare it against our proposed approach using identical neural network architectures, optimizers, initialization schemes, training-validation-test splits, and early stopping procedures.

Simulation design We use the Bermudan max-call option pricing example from Becker et al. [2019], where the price dynamics and reward are given by:

$$S_t^m = s_0^m \exp \left([r - \delta_m - \sigma_m^2/2]t + \sigma_m W_t^m \right), \quad R_t = e^{-rt} \left(\max_{1 \leq m \leq d} S_t^m - K \right)^+,$$

for $m = 1, \dots, d$.

Here, r is the risk-free rate, s_0^m is the initial price, δ_m is the dividend yield, σ_m is the volatility, W is a d -dimensional Brownian motion with zero instantaneous correlation between its components, K is the strike price, R_t is the discounted reward from stopping at time t , and $x^+ = \max\{x, 0\}$. Time is discretized using exercise dates $t_j = jT/H$, $j = 0, \dots, H$.

Following Becker et al. [2019], we set:

$$K = 100, \quad r = 0.05, \quad \sigma_m = 0.2, \quad \delta_m = 0.1, \quad T = 3, \quad H = 9.$$

We report results for $s_0^m = 100$ and $d \in \{100, 200\}$.

As in Becker et al. [2019], we parameterize the stopping policies by feed-forward ReLU neural networks with two hidden layers, $d + 40$ hidden units per layer, Xavier initialization [Glorot and Bengio, 2010], batch normalization [Ioffe and Szegedy, 2015], and the Adam optimizer [Kingma and Ba, 2015]. We use a minibatch size of 64 and early stopping based on validation-set reward. For each run, we synthesize $M \in \{50, 100, \dots, 300\}$ trajectories, using 50% for training, 20% for validation, and the remainder for held-out evaluation.

Results Figure 15 shows the held-out average reward for DOS and our proposed approach, averaged over 5 runs, with varying dataset sizes for $d \in \{100, 200\}$. Our proposed approach achieves higher rewards in all configurations, despite differing from DOS only in the loss function. The gains are largest in data-limited settings where the policy networks are relatively large compared to the dataset size.

Figure 15: Held-out average reward in the Bermudan max-call option pricing simulations.

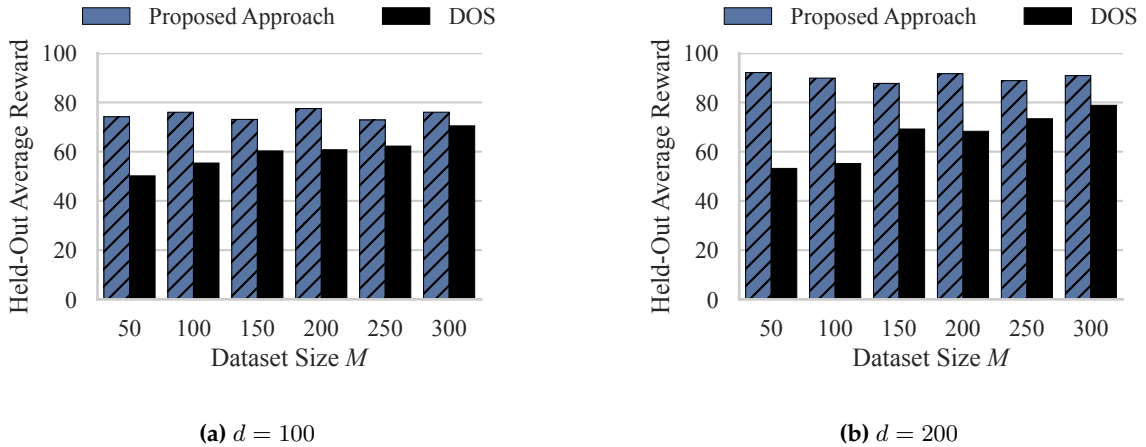
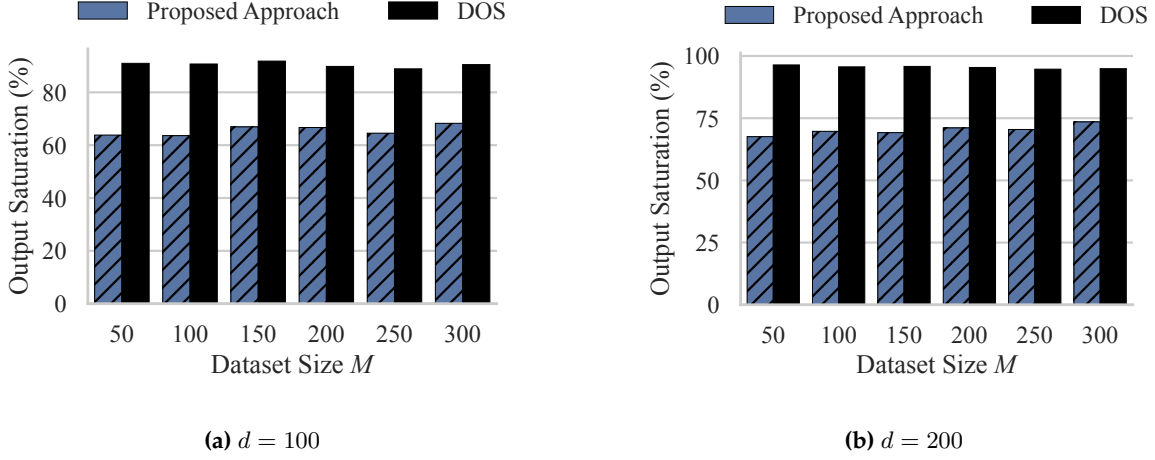


Figure 16 plots the fraction of neural network outputs that become saturated¹⁶ during training (i.e., stopping scores near 0 or 1 indicating confident predictions). DOS exhibits more output saturation than our proposed approach across all configurations. As shown in Appendix B.1, this greater saturation causes gradients to vanish under the DOS loss but not under our proposed loss, which may explain the performance gap.

¹⁶We define a neural network output as saturated if $F_t^{\theta_t}(s) < 10^{-3}$ or $1 - F_t^{\theta_t}(s) < 10^{-3}$.

Figure 16: Output saturation in the Bermudan max-call option pricing simulations.



C Zero-shot LLMs as Policy Models

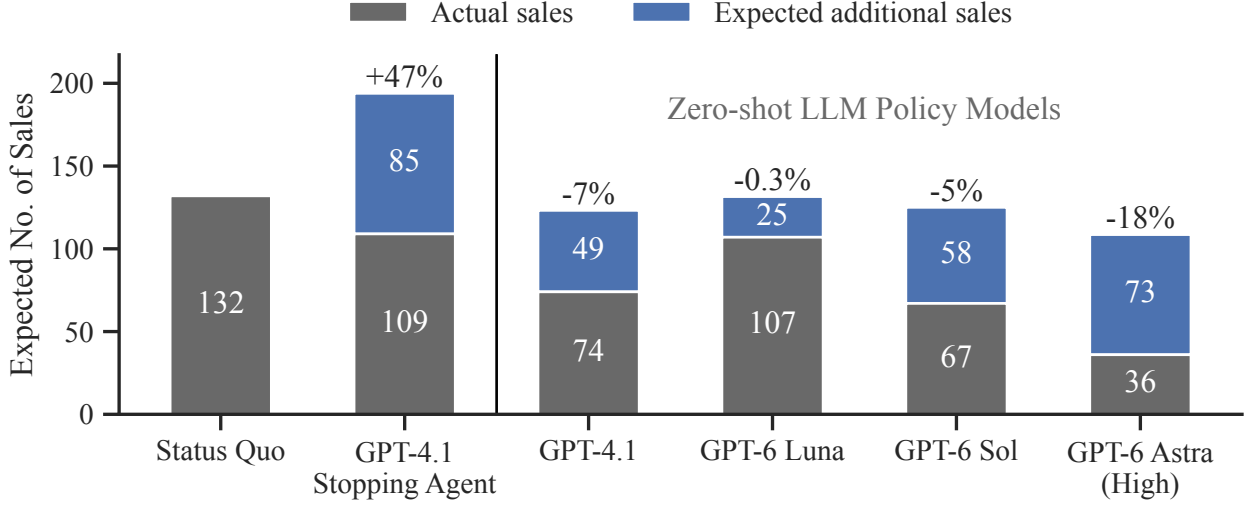
To assess whether LLMs can make effective stopping decisions without task-specific fine-tuning, we evaluate GPT-4.1, GPT-6 Sol, and GPT-6 Astra (with high reasoning). We also evaluate the OpenAI Decisions API, which uses GPT-6 Luna as a decision agent.¹⁷ We use “zero-shot” to mean that the model is not fine-tuned on our task.

At each stopping opportunity, we provide the model with the evolving transcript and use the prompt in Figure 1. For GPT-4.1 and GPT-6 Sol, we use the output probability of the “yes” token as the stopping score. For GPT-6 Luna, the Decisions API returns choice probabilities and a confidence score; we use the probability of the “yes” choice as the stopping score. We threshold the stopping scores from these three models to maximize the cumulative reward on the validation set. The OpenAI API does not provide output-token probabilities for GPT-6 Astra, so we use its output tokens directly as stopping decisions. GPT-6 Astra’s results should therefore be interpreted with the qualification that its decisions cannot be informed by the validation data.

Figure 17 summarizes the results for stopping opportunities at $t \in \{30, 60, 90\}$. The zero-shot GPT-4.1, GPT-6 Sol, GPT-6 Luna, and GPT-6 Astra policies reduce expected sales by 7%, 5%, 0.4% and 18%, respectively. In contrast, our fine-tuned GPT-4.1 stopping agent increases expected sales by 47%. None of the zero-shot benchmarks approaches the performance of our fine-tuned stopping agent. This comparison is consistent with the value of task-specific policy estimation, although the decision rules differ across models, particularly for GPT-6 Astra, whose outputs cannot be informed by the validation set.

¹⁷GPT-6 is OpenAI’s most recent model family; Luna, Sol, and Astra are ordered by increasing capability. GPT-4.1 is not a reasoning model. GPT-6 Sol can be configured to use reasoning, but the OpenAI API does not provide output token probabilities (which we use as stopping scores) when reasoning is enabled. Hence, we report the performance of GPT-6 Sol without reasoning.

Figure 17: Expected sales with our stopping agent and zero-shot LLM policy models, $t \in \{30, 60, 90\}$.



D Differential Performance by Salesperson Effectiveness

We evaluate the returns to our $t \in \{30, 60, 90\}$ stopping agent for salespeople with different levels of effectiveness. We partition the test set into calls made by salespeople with success rates below and above the median salesperson success rate in the test set. Table 2 reports the performance of salespeople and our stopping agent in each subgroup.

Table 2: Expected sales gain for salespeople with success rates below and above the median.

	Low Success Rate Salespeople		High Success Rate Salespeople	
	Salespeople	Stopping Agent	Salespeople	Stopping Agent
No. of Sales (original)	16	13	116	96
Total Time (hours)	51	13	77	29
Additional Sales (expected)	—	27	—	58
Sales Gain (% , expected)	—	150%	—	33%

Low success rate salespeople have a success rate of 1.4% and spend 160 seconds per call on average. High success rate salespeople have a success rate of 9.4% and spend 224 seconds per call on average. We calculate expected additional sales using each salesperson’s time savings, success rate, and average call duration, with the latter two estimated over their test-set calls, and sum these gains within each subgroup.

Our stopping agent increases expected sales by 150% for low-success-rate salespeople and by 33% for high-success-rate salespeople. The gains for low-success-rate salespeople are striking because they rarely succeed and spend about 2.7 minutes per call, so any improvement in efficiency

has a disproportionately large impact. These results suggest that our stopping agent not only boosts overall productivity, but may also reduce performance disparities across salespeople.

E State-of-the-art Offline Reinforcement Learning Methods

We compare our stopping agent with two state-of-the-art methods for stopping policy estimation: deep optimal stopping (DOS) [Becker et al., 2019], designed for feedforward neural network (FFNN) policies, and deep recurrent optimal stopping (DROS) [Venkata and Bhattacharyya, 2023], designed for recurrent neural network (RNN) policies. We train both methods on 3072-dimensional `text-embedding-3-large` embeddings with stopping opportunities at $t \in \{30, 60, 90\}$.

We implement DOS with Xavier-Glorot initialization [Glorot and Bengio, 2010], batch normalization [Ioffe and Szegedy, 2015], and gradient-clipping as recommended by Becker et al. [2019]; these strategies tend to alleviate exploding and vanishing gradients in practice. We sweep over learning rates in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, number of hidden layers in $\{1, 2, 3\}$, and hidden-layer widths in $\{64, 128, 256, 512, 3072\}$, while fixing the batch size at 64, for a total of 60 configurations.

Table 3 (top) shows that the DOS policies perform poorly. The best policy architecture yields expected sales gains of 2%, and the next two best ones produce expected sales losses. With the same embeddings, our stopping agent with a simpler logistic regression policy produces expected sales gains of 3%.

Table 3: Comparing our stopping agent with state-of-the-art offline reinforcement learning methods.

	No. of Sales	Total Time (h)	Additional Sales (expected)	Sales Gain (% expected)
Salespeople	132	128	—	—
DOS				
Policy size = 256×3 , ~ 2.8 M parameters	45	39	89	2%
Policy size = 3072×2 , ~ 56.7 M parameters	48	45	83	-1%
Policy size = 512×2 , ~ 5.5 M parameters	35	33	95	-2%
DROS				
Policy size = 3072×3 , ~ 170.0 M parameters	53	43	84	4%
Policy size = 3072×1 , ~ 56.7 M parameters	45	39	88	1%
Policy size = 256×2 , ~ 3.0 M parameters	42	39	89	-1%

Note: The policy size $H \times D$ indicates H hidden units in each of D hidden layers (FFNN layers for DOS and GRU layers for DROS). For each method, we report the three configurations with the highest validation reward.

For DROS, we implement a PyTorch adaptation of the recurrent optimal stopping policy-gradient (RNN-OSPG) implementation by Venkata and Bhattacharyya [2023] and sweep over the same hyperparameter grid as for DOS. The best policy architecture (Table 3, bottom) yields expected sales gains of 4%, outperforming DOS and our stopping agent with the logistic regression policy, but still underperforming our stopping agents with LLM policies.

F Additional Details for Generalizability

F.1 Evaluating our Stopping Agent on an Out-of-Sample Campaign

We now assess the robustness of our stopping agent to distribution shift by evaluating its performance, *without re-estimation*, on a different outbound sales campaign conducted six months after the original campaign. In this *out-of-sample* campaign, the firm targeted subscribers of a lower-cost sub-brand (analogous to the relationship between Mint Mobile and T-Mobile in the United States) with a similar offer to switch their electricity provider as in the original campaign.

The out-of-sample campaign comprises 8,334 first-contact calls made by 153 salespeople over one month. Only 33 of these salespeople (approximately 20%) also participated in the original campaign. While the average time spent per call was nearly identical across campaigns (195 vs. 196 seconds), the success rate was lower in the out-of-sample campaign (4.9%) compared to the original (5.5%). The difference in targeted subscribers and participating salespeople likely creates differences in the conversational content from our original campaign while maintaining the same sales objective, thereby allowing us to isolate the effect of distribution shift.

The results in Table 4 show that our stopping agent continues to improve sales efficiency in this out-of-sample campaign without re-estimation. Specifically, our stopping agent with stopping opportunities at $t \in \{60, 90\}$ achieves a 15.7% increase in expected sales, while the stopping agent with stopping opportunities at $t \in \{30, 60, 90\}$ achieves a 16.4% increase relative to salespeople, corresponding to 120 and 206 expected additional sales, respectively, despite the temporal gap and change in the subscriber base.

Table 4: Evaluating our stopping agent on an out-of-sample campaign.

	No. of Sales	Total Time (h)	Additional Sales (expected)	Sales Gain (% expected)
Salespeople	405	454	—	—
$t \in \{60, 90\}$ Stopping Agent	348	302	120	15.7%
$t \in \{30, 60, 90\}$ Stopping Agent	266	204	206	16.4%

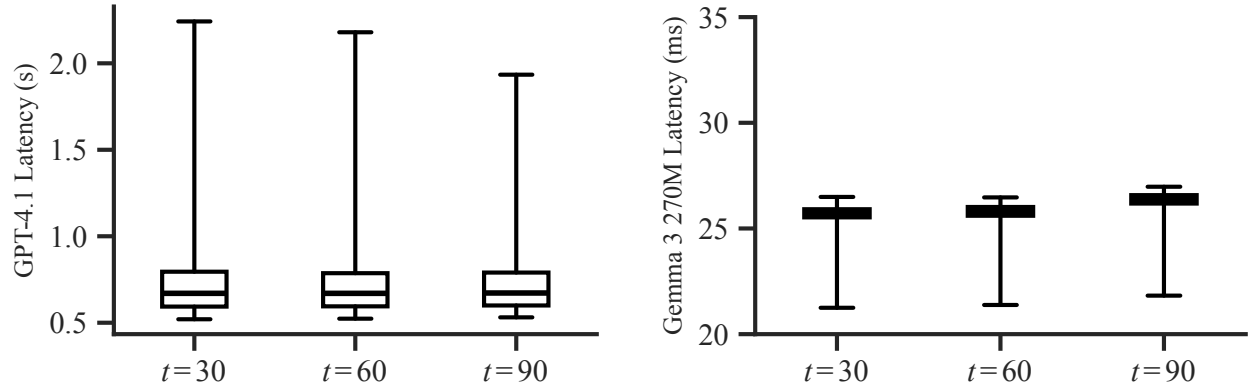
These findings suggest that the conversational patterns and decision thresholds learned by our stopping agent generalize across time and remain robust under (some) distribution shift.

F.2 Measuring Decision Latency

Deploying a stopping agent during live sales calls requires decisions in real time. In Figure 18, we plot the distribution of decision latency for our GPT-4.1 (left) and Gemma 3 270M (right) stopping agents on the test calls at each stopping opportunity $t \in \{30, 60, 90\}$ seconds.

Latency is nearly constant across the three time steps even as the input prompt grows. The GPT-4.1 stopping agent has a median latency of 0.67 seconds per stopping decision, with a 95th

Figure 18: Distribution of stopping decision latency for the GPT-4.1 (left) and Gemma 3 270M (right) stopping agents on the test calls at each stopping opportunity $t \in \{30, 60, 90\}$ seconds.



Note: The horizontal line denotes the median, the box represents the interquartile range, and the whiskers represent the 5th and 95th percentiles of the stopping decision latencies.

percentile latency of roughly 2 seconds. Since GPT-4.1 is an API-based model, this latency includes the network round-trip time between our servers and OpenAI's. The Gemma 3 270M stopping agent, running locally on our server's NVIDIA H100 NVL GPU, has a median latency of just 26 milliseconds per stopping decision and a 95th percentile latency of roughly 27 milliseconds.

For both stopping agents, the median latency stays below the 30-second interval between stopping opportunities, confirming the feasibility of live deployment.