Learning When to Quit in Sales Conversations

Emaad Manzoor,[1*] Eva Ascarza,[2] Oded Netzer,[3]

[1] SC Johnson College of Business, Cornell University
[2] Harvard Business School, Harvard University
[3] Columbia Business School, Columbia University

November 4, 2025

Abstract

Salespeople frequently face the dynamic screening decision of whether to persist in a conversation or abandon it to pursue the next lead. Yet, little is known about how these decisions are made, whether they are efficient, or how to improve them. We study these decisions in the context of high-volume outbound sales where leads are ample, but time is scarce and failure is common. We formalize the dynamic screening decision as an optimal stopping problem and develop a generative language model-based sequential decision agent — a *stopping agent* — that learns whether and when to quit conversations by imitating a retrospectively-inferred optimal stopping policy. Our approach handles high-dimensional textual states, scales to large language models, and works with both open-source and proprietary language models. When applied to calls from a large European telecommunications firm, our stopping agent reduces the time spent on failed calls by 54% while preserving nearly all sales; reallocating the time saved increases expected sales by up to 37%. Upon examining the linguistic cues that drive salespeople's quitting decisions, we find that they tend to overweight a few salient expressions of consumer disinterest and mispredict call failure risk, suggesting cognitive bounds on their ability to make real-time conversational decisions. Our findings highlight the potential of artificial intelligence algorithms to correct cognitively-bounded human decisions and improve salesforce efficiency.

Keywords: salesforce management, telemarketing, 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 Jakurski Family Associate Professor of Business Administration at Harvard Business School. 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, 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 Mosquerra 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 salespeoples' 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, formally known 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. Indeed, in our empirical setting of an outbound sales campaign at a European telecommunications firm, calls that failed to end in a sale lasted almost 3 minutes on average. But just 1 minute into the call, a fine-tuned large language model (LLM) can predict eventual failure to sell quite accurately. Quitting these predictably-risky calls 1 minute in could have saved salespeople 55 hours without sacrificing any sales (further details in Section 4.1).

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 [Shiryaev, 1978], in which the algorithmic decision maker must trade off the immediate benefit of quitting against the option value of continuing to gather information at every instant. Addressing this decision problem requires more than accurate predictions: it requires a principled framework for translating evolving conversational signals into timely, high-stakes actions that maximize cumulative payoffs.

To address this decision problem, we propose a *stopping agent*: a generative artificial intelligence agent that silently observes an ongoing conversation and decides whether and when to quit the conversation in real-time.[1] Although our stopping agent is a generative language model, it never interacts with the prospect. Instead, our stopping agent generates sequential quitting *decisions* conditional on the observed call transcript text to maximize the expected cumulative payoff.

[1] Equivalently, the stopping agent can advise the salesperson to quit, instead of deciding directly.

By parameterizing the stopping policy with a pretrained generative LLM, our stopping agent leverages the state-of-the-art natural language understanding capabilities of such models while also overcoming the inability of dynamic programming to handle high-dimensional states (i.e., call transcripts). However, training language models to become decision policies is challenging, and standard reinforcement learning approaches suffer from training instability, hyperparameter sensitivity, and a lack of computational scalability [Engstrom et al., 2020; Ahmadian et al., 2024].

We address this challenge by formulating the problem of stopping policy estimation as an *imitation learning* problem. We show that optimal quitting decisions can be inferred from historical sales conversations by potentially suboptimal salespeople. We then fine-tune an LLM to generate these inferred optimal decisions given the transcript. In contrast with reinforcement learning approaches, our imitation learning approach is stable, robust to hyperparameters, scales to billion-parameter LLMs, and is compatible with proprietary LLMs only accessible via restrictive APIs.

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 allowed to quit 60 or 90 seconds into a call, our stopping agent retains nearly all sales (130 out of the 132 sales observed in the held-out calls) while reducing the total call time by 36%. Reallocating the time saved to new calls would increase expected sales by 33%. A more 'aggressive' variant of our stopping agent, which is additionally allowed to quit 30 seconds into the call, increases expected sales by 37% by reducing the total call time by 54%. Model training and inference costs totaled only $150, showcasing the cost-effectiveness of our approach.

We further explore systematic patterns in salespeople's quitting decisions and find converging evidence that salespeople are cognitively constrained and struggle to predict eventual call outcomes. Specifically, we find that salespeople seem to under-react to early linguistic indicators of eventual call failure, and that their quitting decisions appear to rely on simple decision rules that overweight a few salient phrases. The phrase most predictive of salespeople quitting by far — *"no me interesa"* ("I'm not interested") — rarely appears early in the call, suggesting that salespeople waste valuable time due to waiting for this salient expression of the prospect's disinterest.

Our stopping agent, in contrast, identifies subtle linguistic indicators of a lack of conversational progress to quit earlier than salespeople. Further, and unlike salespeople, our stopping agent exhibits dynamic variation in its quitting strategy, initially focusing on whether the salesperson is

speaking to the right person, then progressing to whether the prospect is interested, and finally to whether the prospect already has alternatives to the salesperson's offering.

We also examine how salespeople's quitting decisions respond to a plausible shifter of their opportunity cost of time: the proximity of the call to the end of the shift. Calls made near the end of the shift are indeed shorter on average than those made earlier, consistent with higher opportunity costs of time. Under higher time costs, a rational salesperson would selectively shorten calls that are more likely to fail. However, we find that salespeople shorten calls throughout the predicted failure risk distribution, suggesting an inability to accurately predict call failure risk.

Our research makes three key contributions, spanning substantive, methodological, and managerial dimensions. Substantively, we formalize dynamic qualification as an optimal stopping problem and propose a sequential decision-making algorithm using large language models (LLMs) to optimize quitting decisions in live sales conversations. Our approach is the first to enable proactive, data-driven termination of sales calls that balances time costs against sales potential. We show that this approach yields substantial expected sales gains and outperforms several benchmarks.

Methodologically, we develop a stopping agent that functions as a language model-based policy for optimal stopping with high-dimensional textual states. In doing so, we extend the natural language processing literature on detecting adverse conversational outcomes [Zhang et al., 2018] to optimally stopping conversations before such outcomes occur. Our work also presents the first imitation learning solution to optimal stopping with high-dimensional states. Unlike state-of-the-art reinforcement learning approaches [Venkata and Bhattacharyya, 2023], our approach is stable, robust to hyperparameters, and scales to billion-parameter LLMs.

Managerially, we offer a practical and cost-effective approach to improve the efficiency of high-volume outbound sales campaigns. Our stopping agent can be implemented using readily available call transcripts and deployed at a low cost (e.g., $100-$150 per month), even when working with proprietary language models. Moreover, our diagnostic analysis suggests that salespeople are cognitively bounded and struggle to predict call outcomes in real-time, underscoring the need for algorithmic decision support tools to alleviate their cognitive constraints.

The rest of our paper is organized as follows. Section 2 reviews related literature. Section 3 presents our generative language agent for optimal stopping. Section 4 applies our method to call

center sales and evaluates its performance. Section 5 explores drivers of salespeople's suboptimal 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.

The empirical salesforce management literature is primarily focused on mechanisms to motivate selling effort along various dimensions [Misra and Nair, 2011; Chung et al., 2014; Daljord et al., 2016; Kim et al., 2019, 2022; Bommaraju et al., 2025] (we delegate to Misra [2019] for a thorough survey). Our research broadens the scope of salesforce management research by examining how effort can be optimally withheld through disqualification and reallocated to more promising prospects. Rather than designing a salesforce compensation scheme to motivate optimal disqualification, we propose an algorithmic solution to assist salespeople with optimal conversation stopping.

Our work contributes to the growing literature on improving salesperson decision-making using artificial intelligence algorithms. In this literature, Chakraborty et al. [2025] propose algorithms to improve salesforce recruitment, Karlinsky-Shichor and Netzer [2024] propose algorithms to guide pricing, Hu et al. [2024] propose algorithms to match salespeople with the right prospects, and Reeder III et al. [2024] use LLMs to forecast sales revenue from CRM activity logs. Our approach complements this research, and goes beyond prediction to optimize sequential decisions. More broadly, our work contributes to the literature on LLMs as collaborators [Arora et al., 2025].

Methodologically, we propose an imitation learning approach to optimal stopping with high-dimensional textual or conversational states, which enables using language models as policies. Venkata and Bhattacharyya [2023] propose a policy gradients (i.e., reinforcement learning) approach to optimal stopping with high-dimensional non-textual states and recurrent neural network (RNN) policies. As we demonstrate in Section 4, their approach underperforms in our setting.

A broader stream of research applies reinforcement learning to align LLMs with human preferences, including methods such as Proximal Policy Optimization (PPO) [Schulman et al., 2017]

and Group Relative Policy Optimization (GRPO) [Shao et al., 2024]. These approaches assume environments where the language model controls the next state via text generation, whereas the state dynamics in our setting are externally governed by the customer–salesperson interaction. Hence, methods like PPO and GRPO are not directly applicable in our settings.

Our work also relates to the economics and marketing literature on optimal stopping and dynamic discrete choice [Rust, 1987]. Hui et al. [2008] model DVD preorders as optimal stopping decisions, Yoganarasimhan [2013] models bid selection in auctions for freelance projects as optimal stopping rules, and the literature on consumer search models the search process as an optimal stopping problem [Zwick et al., 2003; Branco et al., 2016; Guo, 2022]. More recently, Kang et al. [2025] leverage the equivalence between dynamic discrete choice modeling and inverse reinforcement learning to propose a gradient-based reward function estimation approach, and Barzegary and Yoganarasimhan [2025] propose reducing the state space's dimensionality via recursive partitioning.

The aforementioned literature focuses on estimating dynamic discrete choice models given observed decisions (i.e., estimating structural primitives) and simulating counterfactuals under the assumption that decision-makers behave optimally. In our work, we do not assume that salespeople behave optimally. Instead, we provide a prescriptive (algorithmic) approach to improve potentially-suboptimal stopping decisions. Indeed, we find evidence that salespeople in our setting deviate systematically from optimal stopping behavior.

Our work also relates to the literature on agentic artificial intelligence, wherein LLMs generate actions in addition to natural language. Methods in this literature rely both on careful prompt engineering (e.g., [Bakhtin et al., 2022; Park et al., 2023; Yao et al., 2023; Kim et al., 2023; Yang et al., 2024] and on explicitly training models to act autonomously (e.g., [Chen et al., 2021; Ahn et al., 2022; Schick et al., 2023]). Our work extends this research by building an artificial intelligence agent for optimal conversation stopping and evaluating it on call center sales conversations from the field.

Finally, our work relates to research in behavioral economics using machine learning to evaluate the quality of human decisions [Kleinberg et al., 2018; Mullainathan and Obermeyer, 2022; Rambachan, 2024]. These studies test for screening errors in settings where decision-makers have a one-time choice, such as whether to detain a defendant or administer a medical test, and assess deviations from an implicit threshold rule. We contribute to this literature with an examination of salespeople's dynamic qualification decisions under time pressure.

3 Optimal Conversation Stopping with Generative Language Agents

To support salespeople's dynamic qualification decisions, we develop an algorithmic agent that observes the live conversation and decides whether and when to quit. We build a *language agent*: a language model that generates decisions to optimize a long-term managerial objective. Unlike traditional conversational agents, our language agent does not interact with the prospect. Instead, it observes the conversation between the salesperson and the prospect and optimally quits (or advises quitting) to maximize expected profits. We refer to such language agents as *stopping agents*.

3.1 Problem Definition

Since conversations evolve sequentially, quitting decisions are inherently dynamic. Each moment spent on a call generates information about whether it is likely to end in a sale, but also incurs an opportunity cost: the time could have been allocated to a different prospect. This trade-off motivates formulating qualification as an *optimal stopping problem* with textual states. We adapt the discrete-time finite-horizon optimal stopping formulation of Shiryaev [1978] to our setting.

Formally, at each period $t \in \{1, \ldots, T\}$, the state $s_t \in \mathcal{S}$ consists of the verbatim transcript of the conversation up to period t. A policy $\pi_\theta(a_t|s_t)$, parameterized by θ, observes s_t and selects an action $a_t \in \mathcal{A} = \{\textsf{wait}, \textsf{quit}\}$ at each time t. If the policy chooses wait, it receives a waiting reward w_t, and the process transitions to the next state according to the (unknown) conversational dynamics $\mathcal{P}(s_{t+1}|s_t, a_t)$.[2] If the policy chooses quit, it receives a terminal reward q_t and the process terminates. We impose $a_T = \textsf{quit}$ and set T to the conversation duration (i.e., a dummy terminal period), which enables denoting policies that never quit as policies that quit at T without additional notation.

Let $\tau \in \{1, \ldots, T\}$ denote the stopping time in a conversation induced by a policy π_θ, defined as the earliest time at which the action is quit:

$$\tau = \min\{t \in \{1, \ldots, T\} : a_t = \textsf{quit}\}.$$

[2]Importantly, wait implies the policy *doing nothing*; the conversation proceeds fully unaffected. Formally, $\mathcal{P}(s_{t+1}|s_t, a_t = \textsf{wait}) = \mathcal{P}(s_{t+1}|s_t)$, and $\mathcal{P}(s_{t+1} = \textsf{end state}|s_t, a_t = \textsf{quit}) = 1$. The end state does not require the salesperson to quit immediately, it only implies that no further actions by the stopping policy are permitted, and is compatible with the salesperson ending the conversation naturally. This structure distinguishes optimal stopping problems from general sequential decision problems. Our key innovation is to leverage this distinction for efficient policy estimation (Section 3.3). Note that our stopping policy cannot advise continuation if the salesperson decides to quit, nor can it advise the salesperson what to say; we delegate these extensions of the action space to future work.

The objective is to find a policy π_θ that maximizes the expected cumulative reward $J(\theta)$, defined as:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{\tau-1} w_t + q_\tau\right], \tag{1}$$

where the expectation is taken over the distribution of state-action trajectories induced by π_θ and the transition dynamics $\mathcal{P}(s_{t+1}|s_t, a_t)$ over all calls.

The waiting and quitting rewards w_t and q_τ are exogenously specified by the firm to reflect its operational costs and demand-side conditions. For example (as we show in Section 4), the firm can set w_t based on the opportunity cost of time, such that the cost of waiting equals the expected revenue from spending that time on another call. Similarly, q_τ can be defined as: (a) zero if the stopping agent quits before the salesperson makes a sale, and (b) equal to the profits a successful sale generates if the stopping agent does not quit before the salesperson makes the sale.

3.2 Algorithmic Challenges in Solving the Optimal Stopping Problem

A natural starting point for solving optimal stopping problems is dynamic programming [Bellman, 1966]. In principle, the optimal value functions $V_t^*(s) = \max_\theta \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{\tau-1} w_t + q_\tau \,\middle|\, s_1 = s\right]$ satisfy Bellman's optimality conditions and can be computed recursively via backward induction:

$$V_t^*(s) = \max\left(q_t,\, w_t + \mathbb{E}[V_{t+1}^*(s_{t+1})|s_t = s, a_t = \textsf{wait}]\right), \quad V_T^*(s) = q_T. \tag{2}$$

The optimal policy at time t selects quit if and only if $q_t > w_t + \mathbb{E}[V_{t+1}^*(s_{t+1}) \mid s_t, a_t = \textsf{wait}]$.[3]

While the formulation above is conceptually straightforward, estimating the optimal stopping policy in our setting — conversational sales — presents two key challenges. These stem from the high dimensionality of textual data and the practical issues of reinforcement learning (RL) when applied to estimate large language model (LLM) policies.

Challenge 1: Textual states and the curse of dimensionality. The state at each time t is a growing text sequence (i.e., the transcript of the conversation up to that point). If $\mathcal{V}$ is the vocabulary and L is the maximum transcript length, the size of the state space is of the order $|\mathcal{S}| \approx |\mathcal{V}|^L$. This exponential growth renders dynamic programming computationally infeasible

[3]While one may consider a model-based policy that estimates $\mathbb{E}[V_{t+1}^*(s_{t+1}) \mid s_t, a_t = \textsf{wait}]$, this requires an accurate model of $\mathcal{P}(s_{t+1}|s_t)$, which is challenging to estimate in real-world settings. Our approach does not require $\mathcal{P}(s_{t+1}|s_t)$.

and prevents the value function from being stored or calculated in closed form. This curse of dimensionality also affects other state-enumeration approaches, such as Q-learning [Watkins, 1989].

Challenge 2: Practical issues with deep reinforcement learning of large language model policies. When enumerating a large state space is infeasible, the standard remedy is to estimate a parameterized policy $\pi_\theta(\cdot)$. We therefore parameterize our stopping policy with a pretrained generative LLM to leverage the state-of-the-art natural language understanding abilities of such models [Kaplan et al., 2020]. Estimating policies parameterized by LLMs is, in principle, a natural fit for deep reinforcement learning methods, which have recently been used in several marketing applications ([e.g., Liu, 2023; Ko et al., 2024; Ma et al., 2025]).

In practice, however, coupling deep reinforcement learning with LLMs introduces well-known difficulties[4] [Henderson et al., 2018; Engstrom et al., 2020; Ahmadian et al., 2024], notably: (1) unstable optimization performance that is highly sensitive to hyperparameters and implementation details; and (2) substantially higher computational costs than supervised learning. These practical difficulties have renewed interest in supervised learning-based approaches to policy estimation as simpler, robust, and computationally scalable alternatives [Foster et al., 2024].

3.3 Proposed Method: Imitation Learning to Quit

Given the aforementioned challenges, we propose a solution based on *imitation learning* [Pomerleau, 1988]. Imitation learning is a form of supervised learning, and has been successfully used to train decision-making policies in applications ranging from autonomous helicopters [Abbeel and Ng, 2004] to self-driving cars [Bansal et al., 2018]. In essence, imitation learning trains a policy to mimic the actions of an expert policy (such as a human driver) by learning from a dataset of optimal state-action trajectories generated by the expert.

Formally, given a dataset of state-action trajectories $\mathcal{D} = \{(s_1^i, a_1^i), \ldots, (s_t^i, a_t^i)\}_{i=1}^N$ consisting of state-action pairs from an "expert" policy that maximizes the expected cumulative reward in Equation (1), imitation learning seeks a policy $\pi_{\hat{\theta}}(a|s)$ that minimizes the expected action mismatch:

$$\hat{\theta} = \arg\min_\theta \ \mathbb{E}\big[\mathbb{1}\{\pi_\theta(\cdot|s_t^i) \neq a_t^i\}\big] \,,$$

[4]Recent methods to align LLMs with human preferences [Schulman et al., 2017; Shao et al., 2024; Lambert et al., 2024] assume that the next state is formed by appending the generated token to the previous state. In our setting, this assumption does not hold: the next state is determined by the conversation between the customer and the salesperson.

that is, imitation learning finds a policy that replicates the behavior of this "expert" policy and, in doing so, maximizes the expected cumulative reward.

A key requirement of imitation learning is a dataset of state-action trajectories $\mathcal{D}$ from a policy that maximizes the expected cumulative reward in Equation (1), which might appear prohibitive. In settings such as ours, we cannot assume that salespeople behave optimally (in fact, we show evidence of their suboptimality in Section 5). Our key insight is that *despite* their suboptimality, for the purpose of optimizing the objective in Equation (1), $\mathcal{D}$ can be constructed using historical conversations. Specifically, we infer the cumulative reward-maximizing action a_t for each transcript prefix s_t in a historical conversation by calculating the quitting and waiting rewards given the actual conversation outcome. We then train an LLM to imitate (i.e., by generating) the cumulative reward-maximizing actions given the state. Critically, this reduces the original objective in Equation (1) to the scalable and cost-effective task of fine-tuning an LLM.

Our approach bypasses both the intractability of dynamic programming over high-dimensional textual state spaces (Challenge 1) and the instability and hyperparameter sensitivity of reinforcement learning with LLMs (Challenge 2). Moreover, by framing policy learning as language model fine-tuning, our method inherits the scalability and tooling of modern language model fine-tuning pipelines. Further, our approach is readily compatible with proprietary language models that disallow custom loss functions, and is extensible for use with multi-modal language models.

Figure 1 summarizes our approach. We now describe each of these steps in detail.

3.3.1 Inferring Optimal State-Action Trajectories from Historical Conversations. To construct the dataset $\mathcal{D}$ of optimal state-action trajectories using historical conversations, we calculate, for each historical conversation, the quitting time τ that would have maximized the expected cumulative reward given w_t and q_τ. This induces a state-action trajectory comprised of transcripts and the corresponding optimal actions with respect to Equation (1), thereby representing an expert policy without needing access to optimal quitting decisions from salespeople.

The reason why this works lies in the structure of the optimal stopping problem defined in Section 3.1. For optimal stopping problems, the next state s_{t+1} of a historical conversation given the current state s_t and an action a_t is known: it is either the terminal state if $a_t = \textsf{quit}$, or the conversation transcript until $t + 1$ if $a_t = \textsf{wait}$ (since waiting implies doing nothing). This structure

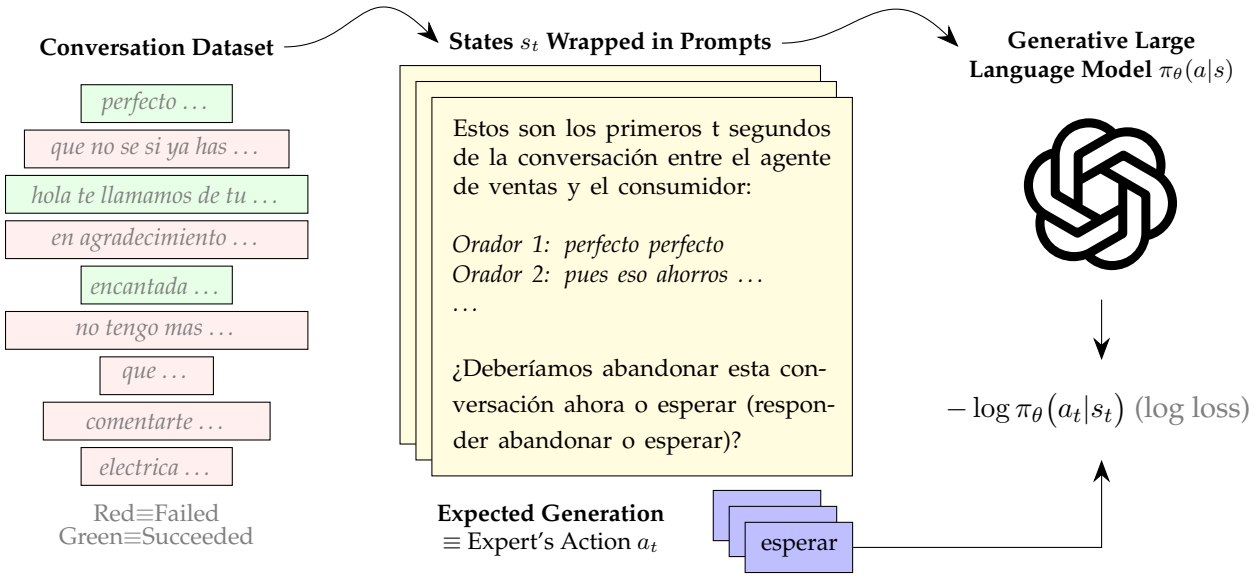


Figure 1: We transform conversations $\mathcal{C}$ into "expert" demonstrations $\mathcal{D}$ (Section 3.3.1), and train an imitation learning policy by fine-tuning an LLM π_θ to generate the "expert's" action a_t for each state s_t wrapped in a prompt (Section 3.3.2). We threshold $\pi_\theta(a|s)$ to obtain deterministic actions (Section 3.3.3).

distinguishes optimal stopping problems from general sequential decision-making problems, and our key innovation is to leverage this structure for imitation learning.

Formally, let $\mathcal{C} = \{(s_1^j, s_2^j, \ldots, s_T^j, y^j)\}_{j=1}^M$ denote a dataset of M conversations, where s_t^j is the transcript of conversation j up to time t, and $y^j \in \{0, 1\}$ indicates whether the conversation ended in a sale. For each conversation j and each candidate quitting time $\tau \in \{1, \ldots, T\}$, we compute the cumulative reward that would have been obtained by quitting at $t = \tau$ as,

$$R_j(\tau) = \sum_{t=1}^{\tau-1} w_t + q_\tau. \tag{3}$$

Let $\tau_j^* = \arg\max_\tau R_j(\tau)$ be the quitting time that maximizes cumulative reward. The corresponding state-action trajectory consists of the transcript prefixes and optimal actions up to time τ_j^*, i.e., $\{(s_t^j, a_t^j)\}_{t=1}^{\tau_j^*}$, where $a_t^j = \textsf{wait}$ for $t < \tau^{*j}$ and $a_{\tau^{*j}}^j = \textsf{quit}$.

We define $\mathcal{D}$ as the union of these optimal state-action trajectories across all conversations:

$$\mathcal{D} = \bigcup_{j=1}^{M} \{(s_t^j, a_t^j)\}_{t=1}^{\tau_j^*}.$$

Data augmentation to enable recovering from suboptimal states. The aforementioned procedure does not include states reached via suboptimal actions in $\mathcal{D}$. For example, if the optimal stopping time is $\tau^* = 2$ for conversation j, $\mathcal{D}$ will include $(s_1^j, \textsf{wait})$ and $(s_2^j, \textsf{quit})$ but will omit (s_t^j, a_t^j) for all $t > \tau^*$. This exclusion poses a risk: if the policy ever reaches a suboptimal state at inference-time (i.e., by waiting when the optimal action was to quit), it may not know how to act in this state.

We address this risk using the data augmentation strategy of Pomerleau [1988] in the context of autonomous vehicle control (i.e., to teach a self-driving vehicle what to do after it drives off the road). Specifically, we augment $\mathcal{D}$ to teach the agent to recover from suboptimal states by adding the suboptimal-state-optimal-action pair $(s_t^j, \textsf{quit})$ for each $t = \tau^* + 1, \ldots, T$ and for each conversation j having optimal stopping time τ^*. That is, for each conversation with an optimal stopping time of $\tau^* < T$, we include in $\mathcal{D}$ the optimal decisions for time periods after τ^*.

3.3.2 Expert Mimicry as Conditional Language Generation. The dataset $\mathcal{D}$ comprises textual states and their corresponding optimal actions (quit or wait). Hence, we train an LLM to mimic the optimal actions by fine-tuning it to *generate* the quit and wait text (i.e., tokens) conditional on the conversation transcript text up to that point. Specifically, we estimate an LLM policy π_θ by minimizing the empirical log-loss of its state-conditioned token generation over $\mathcal{D}$:

$$\hat{\theta} = \arg\min_{\theta} \; \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} \left[-\log \pi_\theta(a_t | s_t) \right]. \tag{4}$$

Thus, we transform the problem of maximizing the expected cumulative reward to that of fine-tuning an LLM to generate the optimal action a_t given the state s_t. s_t may be wrapped in a prompt (our specific prompt is in Figure 1), which can include any desired context (e.g., metadata, images).

3.3.3 From Probabilities to Deterministic Actions: Backward Induction Threshold Tuning. Training an LLM policy by minimizing Equation (4) yields a stochastic policy $\pi_{\hat{\theta}}(a|s)$ that outputs a probability distribution over actions, i.e., $\pi_{\hat{\theta}}(a \mid s) \in [0, 1]$. To implement this policy in practice, we must convert its probabilistic outputs into deterministic decisions. We do so by introducing thresholds $\lambda_t \in [0, 1]$ for each decision point $t = 1, \ldots, T$, such that the policy selects quit if $\pi_{\hat{\theta}}(\textsf{quit} \mid s_t) \geq \lambda_t$ and wait otherwise.

While a grid search over $\lambda_1, \ldots, \lambda_T$ is a straightforward tuning approach, it scales exponentially as $O(B^T)$, where B is the number of grid points. To overcome this, we propose a backward

induction-style threshold tuning procedure that scales linearly in T. The idea is to start at T, where quitting is mandatory, and inductively move backwards. At each time step t, the threshold λ_t is set to maximize the expected reward, accounting for both the immediate reward from quitting and the future reward from waiting at t and quitting later. Algorithm 1 summarizes the procedure.

Algorithm 1 Backward Induction Threshold Tuning

Require: Validation dataset $\mathcal{D} = \{(s_1^i, \ldots, s_T^i, y^i)\}_{i=1}^N$, fine-tuned stochastic policy $\pi_{\hat{\theta}}$, horizon T

$\lambda_T \leftarrow 0$ ▷ Always quit at T
for all $(s_1^i, \ldots, s_T^i, y^i) \in \mathcal{D}$ **do**
  $R_T^i \leftarrow q_T$
end for
for $t = T-1, T-2, \ldots, 1$ **do**
  for all $(s_1^i, \ldots, s_T^i, y^i) \in \mathcal{D}$ **do**
    $R_t^i \leftarrow q_t$ ▷ Reward for quitting at t
    $\tau \leftarrow \min\{u > t : \pi_{\hat{\theta}}(\textsf{quit} \mid s_u^i) \geq \lambda_u\}$ ▷ Next quitting time if agent waits at t
    $R_\tau^i \leftarrow$ reward of quitting at τ ▷ Known by induction for $\tau \in \{t+1, \ldots, T\}$
  end for
  $\lambda_t \leftarrow \arg\max_{\lambda \in [0,1]} \frac{1}{N} \sum_{i=1}^N \left(\mathbb{I}[\pi_{\hat{\theta}}(\textsf{quit} \mid s_t^i) \geq \lambda] R_t^i + \mathbb{I}[\pi_{\hat{\theta}}(\textsf{quit} \mid s_t^i) < \lambda] R_\tau^i \right)$
end for
return $\{\lambda_t\}_{t=1}^T$

3.4 Putting It All Together

We introduce stopping agents — generative language models that make real-time disqualification decisions in conversational sales settings. These agents solve optimal stopping problems over textual state spaces by mimicking an inferred expert policy. Our proposed approach avoids the practical limitations of reinforcement learning by relying instead on imitation learning, enabling stable and cost-effective training through standard language model fine-tuning.

Stopping agents are silent companions to salespeople: they observe the evolving transcript of a sales conversation and decide (or advise the salesperson) whether and when to terminate the call. They are practical to implement and compatible with proprietary language model APIs. They can be easily initialized with visual-, audio-, and multi-modal language models. To facilitate adoption and further research, we provide an open-source framework to build, train, and evaluate stopping agents at `stoppingagents.com`.[5] Having introduced the design and training of our stopping agent, we now apply it to real-world sales data to evaluate its performance.

[5]Note that while stopping agents are trained offline, they are deployed in live conversations and operate in real-time.

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. The dataset used in our analysis comprises 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]. In addition, we observe the salesperson's identifier, the call outcome (i.e., whether the call resulted in a sale based on the consumer confirming the energy contract, hereafter call *success*), and metadata such as the call 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 time spent on calls, with an average duration of 169 seconds. Successful calls were fewer in number but longer on average, reflecting the additional time required to transact and close the sale. The average salesperson made 147 calls of which only 6.6% succeeded. Overall, these patterns suggest that failed calls are significantly time-consuming. Our objective is to reduce the time spent on failed calls without compromising the number of sales.

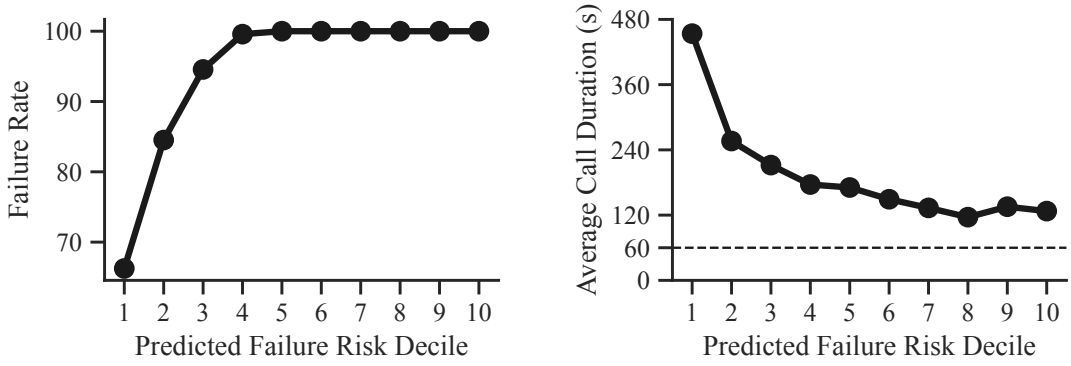
[6]Their compensation consists of a base pay plus a bonus across a two-shift workday with a fixed number of hours. A large number of leads are assigned to each salesperson at random (i.e., salespeople do not choose their prospect pool).

For our subsequent analysis, we randomly partition our dataset into a training set of 5,690 calls (∼50%), a validation set of 3,499 calls (∼30%), and a test (i.e., held-out) set of 2,438 calls (∼20%), stratified to ensure that the proportion of successful calls (∼5.5%) is consistent across all splits.

Motivating evidence. To motivate our work, we build a predictor of call failure risk using the early call transcript. This predictor does not make decisions; it only estimates failure risk at a specified time instant, and does not know when to act on these predictions (in contrast with our stopping agent). We use this predictor to study two questions: (1) do sales call transcripts contain early linguistic indicators of eventual call failure?, and (2) how do salespeople react to them?

Our predictor is the GPT-4.1 [OpenAI, 2025] large language model fine-tuned on the training set calls to predict eventual failure to sell given the first 60 seconds of the call transcript text.[7] We apply this predictor to the test set calls, sort them into deciles of predicted failure risk estimated at $t = 60$, and plot the actual failure rate in each predicted failure risk decile in Figure 2a.

Figure 2: Empirically assessing (a) how well the predicted failure risk correlates with the actual failure rate, and (b) whether and how salespeople react to early indicators of eventual call failure.



(a) **(b)**

Figure 2a shows that the failure risk predicted by GPT-4.1 given the first 60 seconds of the call transcript is quite correlated with the actual failure rate. In fact, the predictor achieves a held-out AUC (which measures predictive performance, where 100% is best) of 94%. Figure 2a thus supports the existence of early indicators of eventual call failure in the call transcript text. In the absence of such indicators, we would expect a lower out-of-sample predictive performance.

[7]Specifically, 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 60 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. We use the probability of generating the 'no' token as the predicted failure risk.

Figure 2b further shows that salespeople spend less time on average on predictably-risky calls. The downward-sloping curve indicates that salespeople are responsive to early indicators of eventual call failure. However, they seem to *under*-react. On calls in the 6 highest predicted failure risk deciles, salespeople spent a total of 55 hours and an average of 2.3 minutes per call, though (as shown in Figure 2a) *all* of these calls ultimately failed to end in a sale. These predictably-risky calls could have been abandoned just 60 seconds in with *no loss* in sales.

The presence of early indicators of eventual call failure, in addition to salespeople's under-reaction to them, motivates our proposed decision support tool to stop sales calls early.

4.2 Training our Stopping Agent

We now describe how to train and evaluate our stopping agent using the dataset in Section 4.1.

Reward configuration. To configure the reward structure of our stopping agent, we specify the waiting and quitting rewards w_t and q_t in Equation (1). We define $w_t = -c$, where $c > 0$ is the opportunity cost per unit time. We set c to reflect the expected value of reallocating the time spent towards initiating new calls and generating new sales, under the assumption that calls are drawn from the same distribution[8] (i.e., with the same success rate and average duration):

$$c = \frac{1}{\text{Average call duration}} \times \text{Success rate} \times \text{Time until next period}.$$

We define the quitting reward as $q_t = b\mathbb{I}[y = 1 \wedge t = T]$, where $y \in \{0, 1\}$ indicates whether the call succeeded, and $b > 0$ denotes the benefit of a sale. We set $b = 1$ so each sale yields a unit reward. Critically, this reward is only realized if the agent chooses not to quit at any point during the call.

Training and evaluating the policy. Following Section 3.3, we train our stopping agent by fine-tuning GPT-4.1[9] on an imitation dataset $\mathcal{D}$ inferred from calls in the training set. We tune the action thresholds on the validation set, and report all performance metrics on the held-out test set.

[8]This is a reasonable assumption in high-volume sales settings where the prospect supply is ample. In other settings, the reward structure can accommodate factors such as lower labor costs, improved salesforce morale, or (as noted by our partner firm) improved customer satisfaction from salespeople spending less time on calls with low-potential prospects.

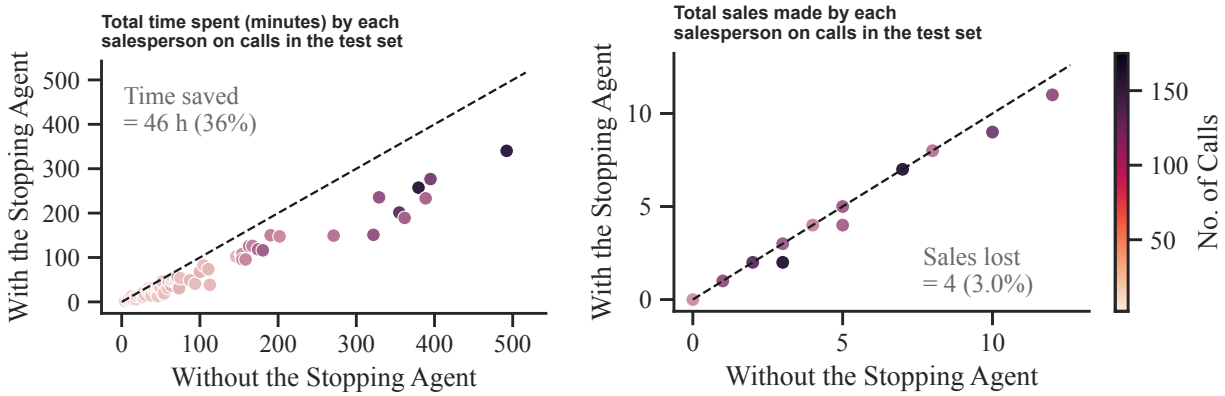
[9]We use the `gpt-4.1-2025-04-14` checkpoint. However, our stopping agent is agnostic to the specific model. In Section 4.5, we demonstrate good performance even with small open-source language models. Our training procedure is identical to that in footnote 7, but uses the prompt and expected response as specified in Figure 1. We found that performance is insensitive to the exact prompt, since fine-tuning tends to eliminate its effect as a prior.

4.3 Evaluating our Stopping Agent's Performance

We start by evaluating a simple stopping agent with $T = 2$ decision opportunities (ignoring the dummy terminal time period) at $t = 60$ and $t = 90$ seconds. We will later increase T to examine the value of more frequent decision-making.

We begin by comparing the *total time spent* and *total number of sales* by each salesperson with and without our stopping agent, as shown in Figure 3. 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 3: 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 $T = 2$ decision opportunities, at $t = 60$ and at $t = 90$.



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 a total of 128 hours on calls in the test set. Our stopping agent reduces call time by 1.37% to 71.6% per salesperson, totaling 46 hours across all salespeople (a 36% reduction in total call time). This reduction comes at the cost of 4 sales, or 3% of the 132 sales in the test set.

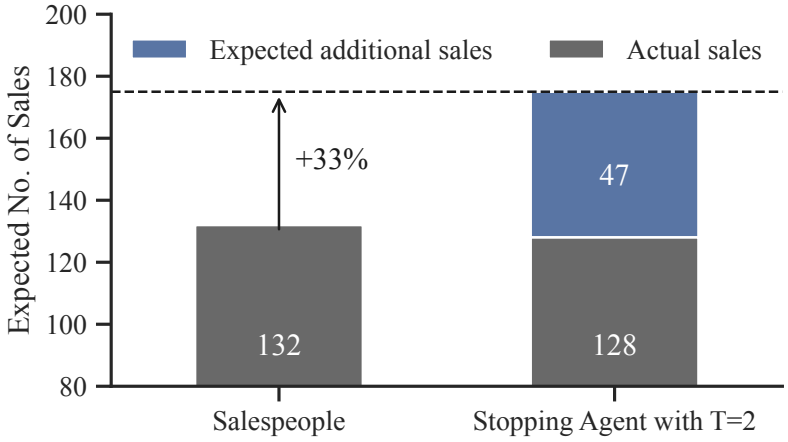
To quantify the net gains from quitting earlier than salespeople, we construct a unified metric that combines time savings and observed sales. Specifically, we measure the *expected* number of sales assuming that the time saved by quitting early is reallocated to initiating new calls:

$$\text{Expected sales} = \text{Actual sales} + \frac{\text{Time saved}}{\text{Average call duration}} \times \text{Success rate}, \tag{5}$$

As in footnote 8, this calculation assumes an ample supply of prospects drawn from the same distribution as the original calls (i.e., with the same average call duration and success rate).

We visualize the actual and expected number of sales on the test set calls for salespeople and for our stopping agent in Figure 4. With only $T = 2$ decision opportunities at $t = 60$ and $t = 90$ seconds, our stopping agent delivers a 33% increase in expected sales (from 132 to 175 sales). Of these, 128 are from the original calls and 47 result from reallocating the 46 hours saved to new calls. These results highlight that even a conservative stopping agent, with only two decision points relatively late in the call, can yield substantial gains in sales effectiveness.

Figure 4: Expected number of sales by our stopping agent with $T = 2$ decision opportunities at $t \in \{60, 90\}$.



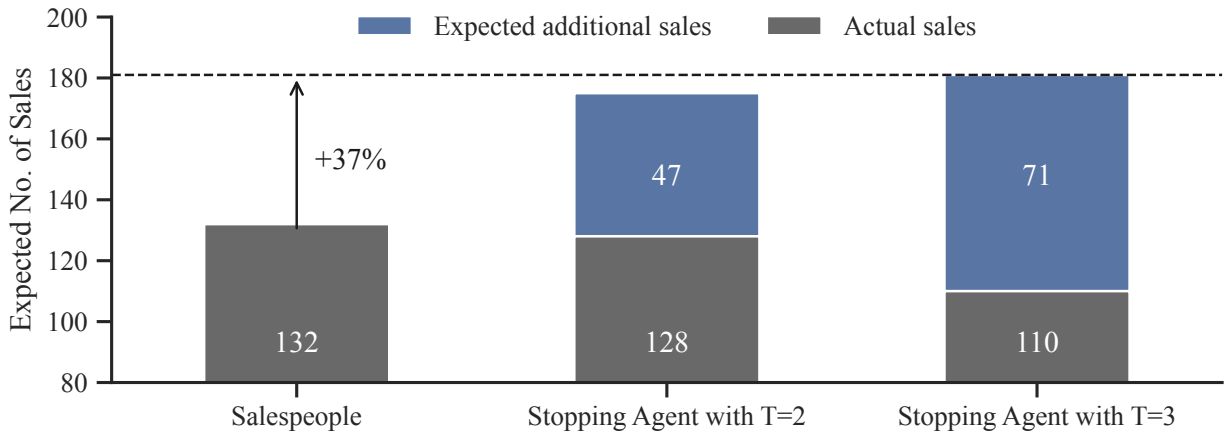
Note: **The expected additional sales is the number of sales expected by reallocating the time saved (46 hours or 36% saved) to making new calls drawn from the same distribution (i.e., with the same average duration and success rate).**

Extending the Decision Horizon. This strong performance naturally raises the question of whether quitting earlier could unlock even greater value. To explore this, we introduce a third decision point at $t = 30$, retrain our stopping agent, and reevaluate its performance. As shown in Figure 5, our stopping agent with $T = 3$ decision opportunities at $t = 30$, $t = 60$, and $t = 90$ achieves a 37% gain in expected sales by reducing the time spent on calls by 69 hours (i.e., 54%).

We continue using $T = 3$ decision points at $t \in \{30, 60, 90\}$ for subsequent analyses, though our proposed approach can be extended to more decision opportunities.[10] We next explore heterogeneity with salesperson effectiveness and benchmark alternative stopping policies.

[10] In practice, the number of decision 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.

Figure 5: Expected number of sales by our stopping agent with $T = 2$ and $T = 3$ decision opportunities.



Differential Performance by Salesperson Effectiveness. We now evaluate the returns to our $T = 3$ 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. 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	97
Total Time (hours)	77	38	51	21
Additional Sales (expected)	—	12	—	45
Sales Gain (%, expected)	—	56%	—	22%

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 use these subgroup-specific averages (instead of the overall sample averages) to translate our stopping agent's time savings to the expected additional sales in each subgroup.

Our stopping agent increases expected sales by 56% for low-success-rate salespeople and by 22% for high-success-rate salespeople. The gains for low-success-rate salespeople are striking because they rarely succeed and spend about 2.5 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.

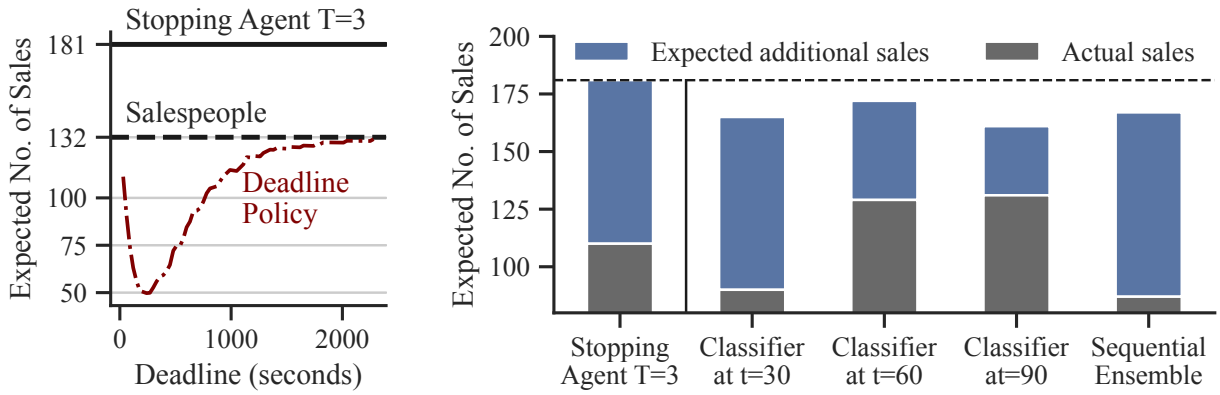
4.4 Evaluating Alternative Stopping Policies

In this section, we compare our stopping agent against several alternative policies, ranging from simple decision rules to state-of-the-art reinforcement learning benchmarks.

4.4.1 Simpler Decision Rules. We first compare our stopping agent with simple policies that either ignore the conversation content or do not account for time costs.

A deadline policy. We begin with a basic *deadline policy* that quits at a fixed time t if the call has not yet succeeded by t. This policy does not consider call content. Figure 6a presents the expected number of sales across different deadlines, alongside the performance of both salespeople and our $T = 3$ stopping agent. Across all values of t, the deadline policy underperforms both salespeople and our stopping agent. This suggests that a simple rule advising salespeople to quit if a call lasts more than t seconds is unlikely to improve sales effectiveness.

Figure 6: Expected number of sales by our stopping agent vs. several content and cost-agnostic baselines.



(a) Expected no. of sales by our $T = 3$ stopping agent and a deadline policy.

(b) Expected number of sales by our $T = 3$ stopping agent, classification-based policies at $t = 30$, $t = 60$, and $t = 90$, and a sequential ensemble.

Classifier-based policies. Next, we assess a set of classifier-based policies. Each policy consists of a GPT-4.1 language model fine-tuned to classify whether the call will eventually succeed using only the first $t = 30$, $t = 60$, or $t = 90$ seconds of the transcript, respectively (similar to the predictor in Section 4.1). At each t, the policy quits if the predicted success probability falls below a threshold selected to maximize balanced classification accuracy on the validation set. As shown in Figure 6b, although the classifier-based policies outperform salespeople, they perform worse than our stopping agent despite being built on the same GPT-4.1 large language model.

A sequential classifier ensemble. A possibly stronger benchmark is a sequential classifier ensemble, which 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 multiple decision opportunities like our stopping agent, but without explicitly incorporating forward-looking cumulative reward maximization. As shown in Figure 6b, this ensemble also underperforms our stopping agent.

In summary, these results underscore the limitations of call content and time cost-agnostic stopping policies. Even when built on the same (GPT-4.1) LLM, such policies fail to match the performance of our stopping agent, which is explicitly trained to make sequential decisions using the call content that maximize the expected cumulative reward.

4.4.2 A State-of-the-Art Reinforcement Learning Method. We benchmark our stopping agent against a state-of-the-art deep reinforcement learning approach for optimal stopping: the optimal stopping policy gradients (OSPG) algorithm [Venkata and Bhattacharyya, 2023]. OSPG is designed for recurrent neural network (RNN) policies, and not LLM policies. Hence, we equip the RNN policies with linguistic knowledge by representing the transcript at $t \in \{30, 60, 90\}$ with 3072-dimensional OpenAI `text-embedding-3-large` vectors. We train policies with $T = 3$ decision points and report the best-performing hyperparameter configurations in Table 3.

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

OSPG (Reinforcement Learning), $T = 3$	No. of Sales	Total Time (h)	Additional Sales (expected)	Sales Gain (%, expected)
Salespeople	132	128	—	—
Policy size = 3092 $\times$ 1, $\sim$10M parameters	117	109	20	4%
Policy size = 3092 $\times$ 5, $\sim$50M parameters	95	99	29	-6%
Policy size = 3092 $\times$ 10, $\sim$100M parameters	109	117	11	-9%

Note: **The policy size $H \times D$ indicates H hidden units in each of D hidden layers. Using the implementation by Venkata and Bhattacharyya [2023], we: (i) fix H to the embedding dimensionality plus 20 (i.e., $H = 3072 + 20$) and vary D to control the total number of parameters, (ii) tune the learning rate in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, and (iii) train for 100 epochs using the AdamW optimizer with early stopping on the validation set.**

All OSPG policies underperform our stopping agent, and the best OSPG policy achieves an expected sales gain of 4%. In fact, the OSPG policies with $\sim$50M and $\sim$100M parameters also underperform salepeople and produce expected sales losses. We also find that many training runs

fail due to the reward plateauing or collapsing to zero, consistent with the documented stability issues of deep reinforcement learning [Engstrom et al., 2020].

Taken together, these results highlight the limitations of both simpler heuristics that ignore the call text and time costs, and more complex reinforcement learning approaches that are not stable or scalable. They reinforce the effectiveness of imitation learning for optimizing sequential stopping decisions with high-dimensional states and externally governed state transitions. Next, we evaluate the versatility of our stopping agent by instantiating it with open-source language models.

4.5 Parameterizing our Stopping Agent with Open-Source Language Models

Unlike proprietary language models, open-source language models can be hosted by firms on-premise, which allows for greater control and privacy. However, open-source language models often exhibit lower performance in terms of predictive accuracy and language understanding relative to proprietary alternatives. Given this trade-off, we next assess whether our stopping agent can still deliver efficiency gains when implemented using open-source language models.

We instantiate our stopping agent with two open-source language models: (i) Gemma 3 with 270 million parameters [Gemma Team et al., 2025], and (ii) Llama 3.2 with 3 billion parameters [Touvron et al., 2023]. Both models are significantly smaller than GPT-4.1 (believed to have trillions of parameters), and thus cost-effective to host on-premise. Our results are in Table 4.

Table 4: Evaluating our stopping agent with open-source language models as the policy.

	No. of Sales	Total Time (h)	Additional Sales (expected)	Sales Gain (%, expected)
Salespeople	132	128	—	—
Gemma 3 (270 million parameters)				
$T = 2$ Stopping Agent	103	95	34	4%
$T = 3$ Stopping Agent	81	65	65	10%
Llama 3.2 (3 billion parameters)				
$T = 2$ Stopping Agent	115	100	28	9%
$T = 3$ Stopping Agent	86	63	65	16%

Even with significantly smaller open-source language models parameterizing the stopping policy, our $T = 3$ stopping agent achieves expected sales gains of 10% to 16% over salespeople. These results show that, even under privacy and computational constraints, firms can effectively deploy our stopping agent to improve sales efficiency.

4.6 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 retraining*, 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 31 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 5 show that our stopping agent continues to improve sales efficiency in this out-of-sample campaign without retraining. Specifically, it achieves a 10% increase in expected sales at $T = 2$ and a 16% increase at $T = 3$ relative to salespeople, corresponding to 39 and 63 expected additional sales, respectively, despite the temporal gap and change in the subscriber base.

Table 5: 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	—	—
$T = 2$ Stopping Agent	328	323	116	10%
$T = 3$ Stopping Agent	275	237	193	16%

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.

5 Diagnosing Salespeople's Quitting Decisions

Having demonstrated that our stopping agent delivers substantial performance gains, we now diagnose why salespeople's quitting decisions fall short, using our stopping agent's quitting deci-

sions 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, in Section 5.1, we train and analyze interpretable machine learning predictors of salespeople's quitting decisions. Our analysis reveals that salespeople decide when to quit using simple decision rules that overweight a few salient phrases that often occur late in the call, whereas our stopping agent uses a more complex and dynamic set of linguistic cues. In Section 5.2, we examine how salespeople's quitting behavior changes under heightened time pressure. We find evidence that salespeople do not sufficiently adjust their quitting decisions to account for call failure risk when the opportunity cost of time is higher.

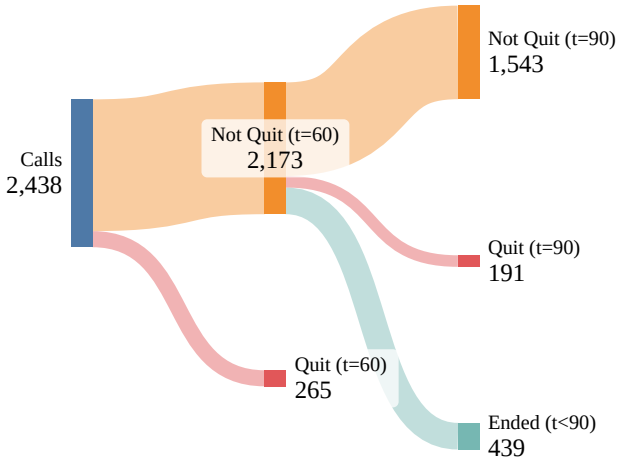
Together, our results suggest that salespeople face cognitive limits when attempting to disqualify consumers early in a call, plausibly due to having to engage with the consumer, address objections, and perform other selling tasks simultaneously. This highlights the value of algorithmic support in reducing cognitive load and improving real-time disqualification decisions.

5.1 Explaining Quitting Decisions using Interpretable Machine Learning

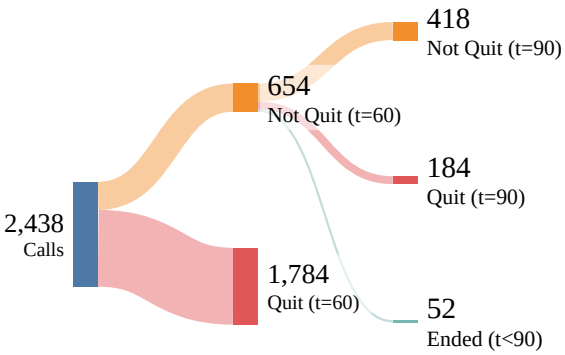
5.1.1 Examining quitting times. We begin by visually examining when salespeople and our stopping agents quit calls in the test set. To do so, we use a series of Sankey diagrams that depict quitting decisions at discrete time points. To enable direct comparisons with our stopping agents, we encode the salesperson's quitting decision at $t \in \{30, 60, 90\}$ as a binary indicator that equals 1 if the call lasts longer than $t + \Delta$ seconds, and 0 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 7 visualizes quitting behavior using Sankey diagrams for (a) salespeople, (b) our $T = 2$ stopping agent, and (c) our $T = 3$ stopping agent, respectively. Figure 7a shows that salespeople hesitate to quit early: they quit *no* calls at $t = 30$, just 11% of calls at $= 60$. and just 11% of the remaining calls $t = 90$. In contrast, our $T = 2$ stopping agent (Figure 7b) quits 75% of calls at $t = 60$, and 31% of the remaining calls at $t = 90$. Our $T = 3$ stopping agent (Figure 7c) quits even more aggressively. These patterns highlight a key behavioral difference between salespeople and our stopping agents: while our stopping agents favor quitting early, salespeople hesitate and delay.

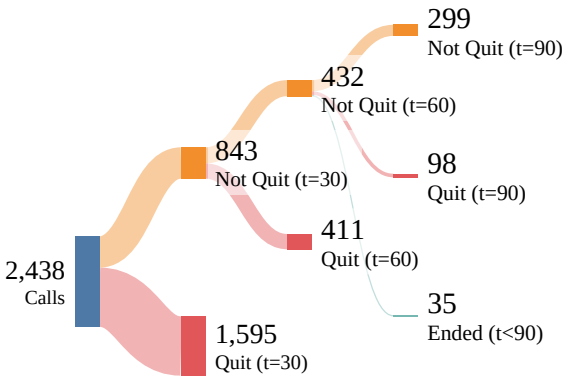
Figure 7: Sankey diagram of quitting decisions by salespeople and by our stopping agents.



(a) Salespeople



(b) $T = 2$ Stopping Agent



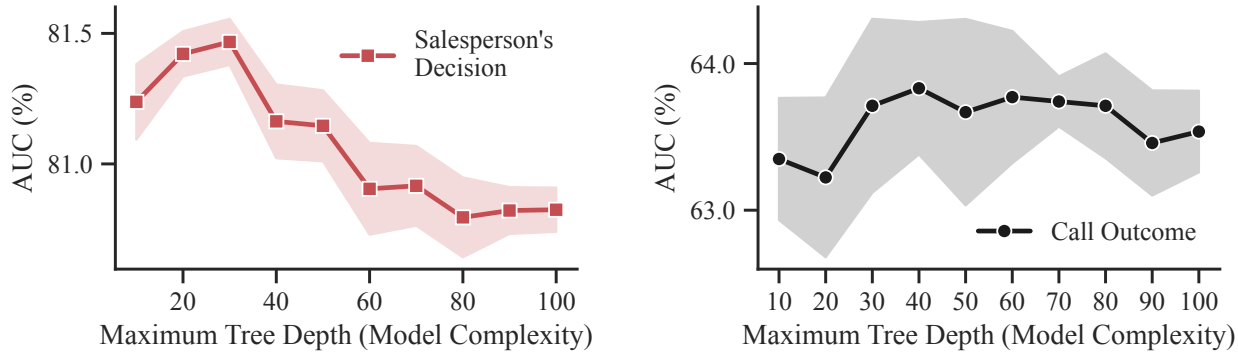
(c) $T = 3$ Stopping Agent

***Note:* Salespeople did not quit any calls at $t = 30$, so we omit that decision opportunity from the Sankey diagram. 439 calls in the test set that were not quit by salespeople at $t = 60$ (by our definition) ended before $t = 90$.**

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. 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 8: 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).**

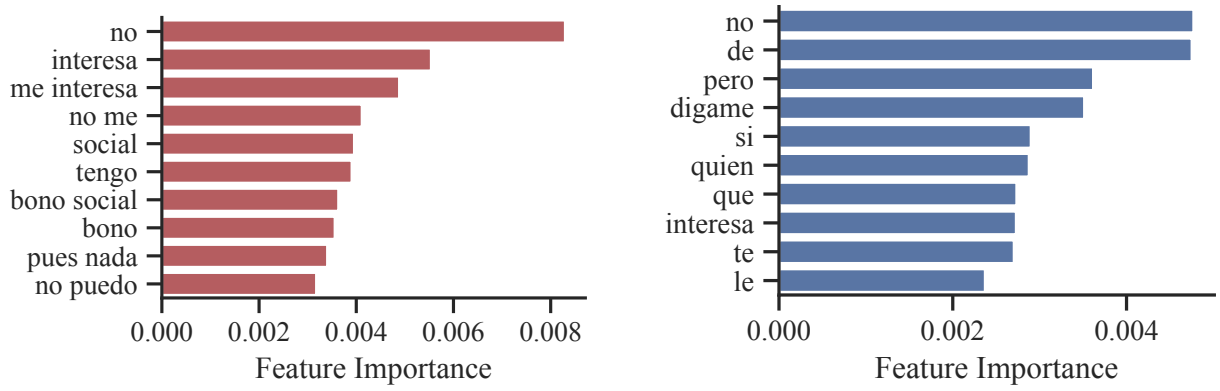
Figure 8 (left) shows the held-out AUC of predicting salespeople's quitting decisions with increasing model complexity. For comparison, Figure 8 (right) shows the held-out AUC of predicting eventual call outcomes given the same features and model. These figures reveal two key insights.

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 9a displays the 10 most predictive features.

Figure 9: Most important features (by Gini importance [Breiman et al., 2017]) in the best random forest model predicting quitting decisions at $t = 60$ by (a) salespeople, and (b) our $T = 2$ stopping agent.



(a) **(b)**

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 $T = 2$ stopping agent's quitting decisions at $t = 60$ in Figure 9b.[11]. We note two observations. First, the distribution of feature importances is less skewed than in Figure 9a. 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

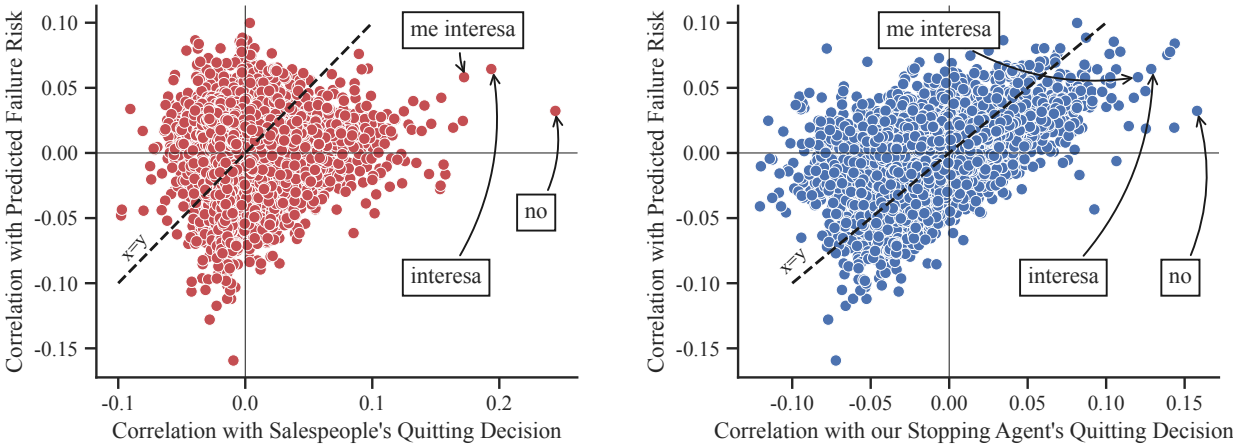
[11]Note that this random forest predictor is trained to predict our $T = 2$ stopping agent's quitting decisions at $t = 60$ on calls in the *validation set*, since our stopping agent was itself trained on calls in the training set.

affirmations of interest or to curiosity about the product. These phrases are more subtle indicators of disinterest than *"no me interesa"*, but are prevalent 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 as in Figure 2b).

Figure 10a 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. Hence, Figure 10a suggests that *"no me interesa"* indeed has an outsized influence on salespeople's quitting decisions relative to other phrases and relative to its association with predicted call failure risk.

Figure 10: 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 $T = 2$ stopping agent.

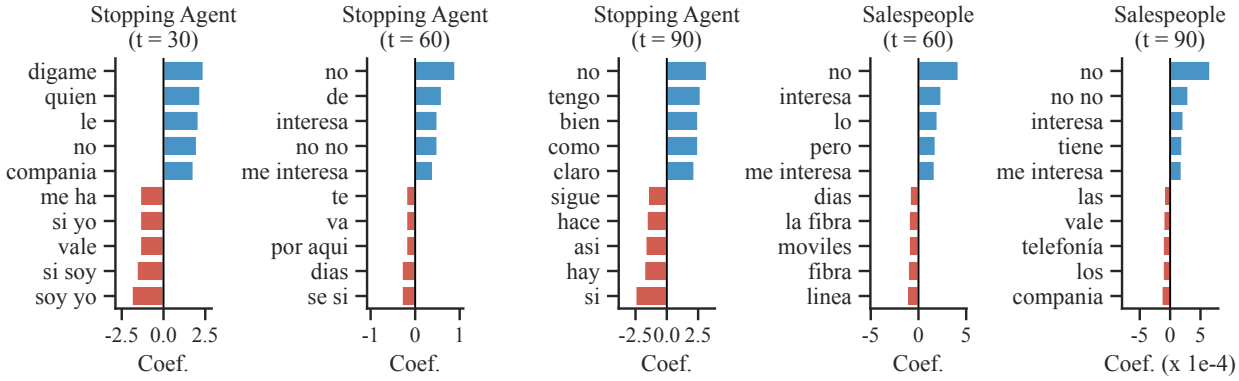


In Figure 10b, we replicate Figure 10a for our $T = 2$ stopping agent. In contrast with Figure 10a, we find that the phrases comprising *"'no me interesa"* do not appear to have an outsized association with quitting relative to other phrases and to their association with predicted call failure risk. Their associations are in line with the other phrases and relatively closer to the $y = x$ line than in Figure 10a. This further suggests that our stopping agent is less biased by the salient *"no me interesa"* phrase, and relies on a wider range 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 models[12] of our $T = 3$ 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 10, this analysis measures *conditional* correlations of each unigram and bigram with the quitting decision at t. Further, it allows measuring the direction or sign of each association (unlike the feature importances in Figure 9), 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 11.

Figure 11: 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 11 reveals several insights. Consistent with Figure 9a, 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 decision opportunities. In contrast, the linguistic cues used by our stopping agent exhibit dynamic variation as the call progresses. Notably:

[12]We 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).

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 $t = 90$ the stopping agent quits based on the *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

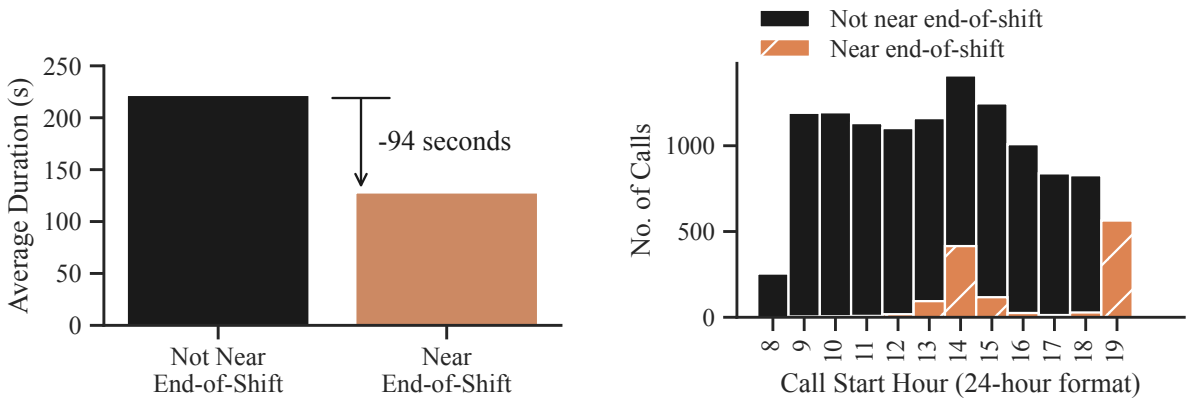
We previously used interpretable machine learning to diagnose deficiencies in salespeople's quitting decisions. Previous research shows that decision-making deficiencies are more pronounced during periods of high fatigue or pressure (e.g., Danziger et al. [2011] find that judges are less likely to grant 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.

Intuitively, when time becomes more costly, we would expect salespeople to reallocate effort toward higher-value opportunities. In particular, they should be more likely to shorten calls with a high risk of failure, and preserve time for those with a low risk of failure. Such behavior would indicate risk-sensitive time management. However, if salespeople do not shorten high-risk calls more than low-risk ones when time is scarce, it suggests that salespeople mispredict whether a call will fail or are insensitive to call failure risk.

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 eager to finish on time, avoid starting long conversations, or meet their 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 12a 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 12b 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 12: Whether a call was made near the end of the salesperson's shift as a time cost-shifter.



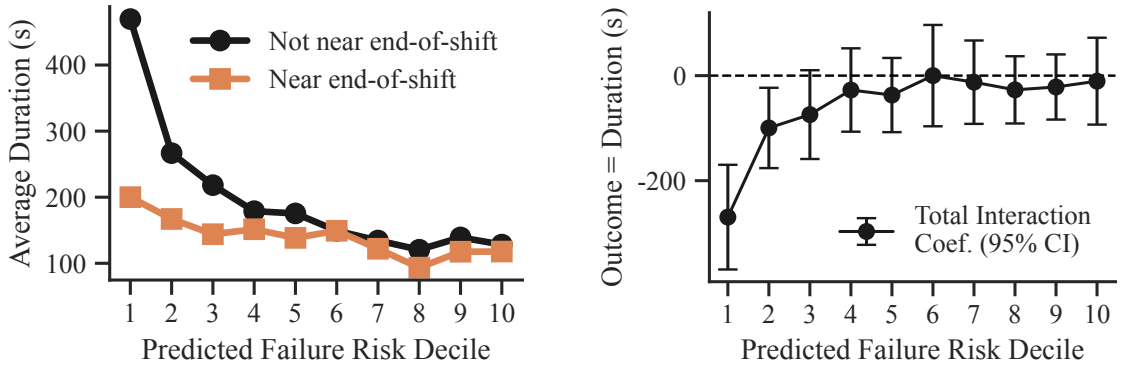
(a) Avg. duration of "near end-of-shift" calls vs. others. **(b) Start times of "near end-of-shift" calls vs. others.**

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 GPT-4.1 as in Figure 2b). Figure 13a 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 lesser 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 13a 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 13: 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.



(a) Call duration by predicted failure risk

(b) Total interaction effect coefficients

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:

$$\text{Duration}_i = \alpha_0 + \gamma_0 s_i + \sum_{j=2}^{10} \alpha_j p_{ij} + \sum_{j=2}^{10} \beta_j p_{ij} s_i + \epsilon_i, \tag{6}$$

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 13b 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.

These results, taken together with our earlier findings, point to a broader failure to integrate risk information into disqualification decisions, even when opportunity costs are elevated. This behavior may stem from salespeople's limited ability to accurately assess call failure risk.[13]

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, do not adjust the cues they rely on as the call progresses, and do not adjust their quitting behavior in response to elevated opportunity costs of time. When faced with higher opportunity costs of time near the end of their shift, salespeople shorten calls indiscriminately, including those with strong sales potential.

Together, these patterns suggest that suboptimal quitting arises due to cognitive bounds: a limited ability to integrate failure risk, time costs, and conversational context into real-time decisions. This reinforces the need for decision-support tools like our stopping agents that systematically incorporate these factors to guide real-time quitting 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

[13] In Appendix A, we supplement this analysis with a contraction test [Kleinberg et al., 2018], which examines whether salespeople misrank calls by their failure risk under the assumption that selling is purely informative. The contraction test compares salespeople with an imperfect algorithm that predicts call outcomes at $t = 60$, and reveals that 80% of the salespeople in our dataset misrank calls by their failure risk.

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 imitation 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 stopping agent reduces the time spent on failed calls by 54% while preserving nearly all sales. Reallocating the time saved to new calls increases expected sales by up to 37%. Applying our stopping agent to an out-of-sample campaign that was not used for training also produces a substantial expected sales gain of 16%, suggesting that the performance of our stopping agent is durable and robust to delayed retraining.

Notably, our stopping agent delivers these gains using only the transcript text (without visuals or voice), and can be trained and deployed cost-effectively using either commercial APIs or open-source models. 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 companies 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

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

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 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 the 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 imitation 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 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 imitation 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

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. 2004.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267. 2024.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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 Baevski, 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.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. doi:10.1126/science.ade9097.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Ebrahim Barzegary and Hema Yoganarasimhan. A recursive partitioning approach for dynamic discrete choice modeling in high dimensional settings. *arXiv preprint arXiv:2208.01476v2*, 2025.

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.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

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.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation Matters in Deep RL: A Case Study on PPO and TRPO. In *International Conference on Learning Representations*. 2020.

Dylan J Foster, Adam Block, and Dipendra Misra. Is Behavior Cloning All You Need? Understanding Horizon in Imitation Learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

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.

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.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32. 2018.

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.

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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36:39648–39677, 2023.

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.

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, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Xiao Liu. Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Marketing Science*, 42(4):637–658, 2023.

Liangzong Ma, Ta-Wei Huang, Eva Ascarza, and Ayelet Israeli. Dynamic Personalization with Multiple Customer Signals: Multi-Response State Representation in Reinforcement Learning. *Available at SSRN*, 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.

OpenAI. Introducing GPT-4.1 in the API. 2025. Accessed: 2025-05-01.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22. 2023.

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.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

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, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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.

CJCH Watkins. Learning from Delayed Rewards. *PhD Thesis, Cambridge University, Cambridge, England*, 1989.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-Agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*. 2023.

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.

Justine Zhang, Jonathan P Chang, Lucas Dixon, Yiqing Hua, Nithum Tahin, and Dario Taraborelli. Conversations Gone Awry: Detecting Early Signs of Conversational Failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1. 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.

Appendix A: A Contraction Test of Salespeople's Quitting Decisions

Kleinberg et al. [2018] introduce *contraction* as a nonparametric empirical test to detect potential misranking by human decision-makers in settings where counterfactual outcomes are observed for some but not all decisions. For example, whether a defendent commits a crime before trial is observable if the judge releases them, but not if they are detained.

The key insight behind the test is that even when we cannot observe the outcomes of some decisions, we can assess humans' ranking quality by comparing their decisions to those that would have been made by a predictive algorithm. The contraction test simulates replacing human decisions with algorithmic ones in order of the algorithm's predicted risk or value—*contracting* the set of positive human decisions. For example, when testing judges for misranking, Kleinberg et al. [2018] simulate the release of defendants in order of their algorithm-predicted risk of reoffending.

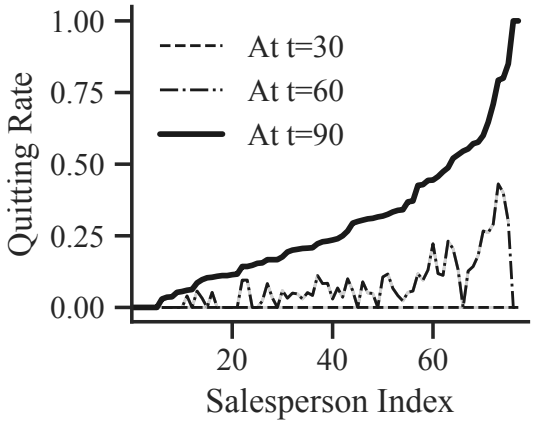
This test relies on the assumption that the decision does not itself affect the counterfactual outcome, only who is selected into the decision. For example, in Kleinberg et al. [2018], a judge releasing or detaining a defendant does not change the defendant's underlying propensity to commit a crime; the decision to release only *reveals* but does not affect the crime risk. Under this assumption, one can compare outcomes across different decision-makers with varying thresholds (e.g., stricter versus more lenient judges) to assess whether the ranking of released individuals implied by judges decisions is consistent with outcome risk.

We apply the contraction test to quitting decisions at time t, analogous to a judge's decision to jail a defendant. We assume that the salesperson's decision to not quit the call at t has no causal effect on the likelihood of a sale, it simply allows the outcome to reveal itself. This assumption allows us to treat variation in salespeople's behavior (e.g., some salespeople persist more than others) as a source of quasi-random variation in treatment assignment, and thus to compare outcomes across salespeople with different levels of persistence. Applying the contraction test in this setting allows us to assess whether salespeople are effectively identifying and prioritizing high-value leads, or whether an algorithm trained to predict conversion would produce better-ranked decisions.

We operationalize quitting at t as a binary indicator of whether the call ends before $t + \Delta$ seconds, where $\Delta = 10$ reflects the average duration of closing salutations. The contraction test requires variation in quitting behavior across individuals. Figure 14 shows each salesperson's

quitting rate at $t = 30$, $t = 60$, and $t = 90$ seconds. Because quitting rates exhibit limited variation at $t = 30$ and $t = 60$, we focus on $t = 90$, where there is sufficient heterogeneity to apply the test.

Figure 14: Quitting rate of each salesperson at $t = 30$, $t = 60$, and $t = 90$ seconds.

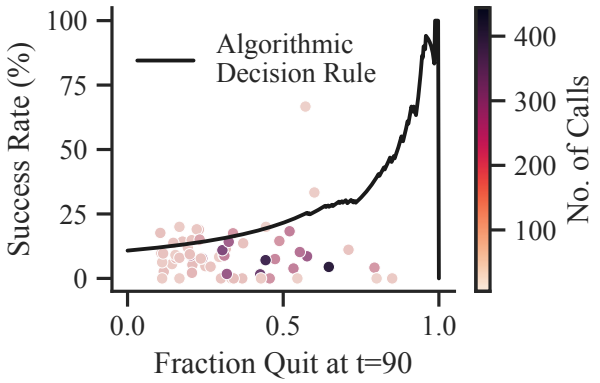


To construct the algorithmic decision rule, we simulate quitting calls from the test set made by the most lenient salespeople, those in the first quintile of the quitting rates distribution at $t = 90$, in decreasing order of predicted failure risk. Predictions are generated by a fine-tuned GPT-4.1 model given the call transcript up to $t = 90$. Figure 15 plots the resulting algorithmic decision rule curve. At a quitting rate of 0%, the curve reproduces the observed success rate of lenient salespeople (10.8%). At 100%, the success rate falls to 0%. Between these extremes, the success rate of the algorithmic decision rule increases as predictably-risky calls are removed, though not monotonically since the algorithm is imperfect and may misrank calls.

The contraction test then compares this curve to human decision-making. For each non-lenient salesperson (quintiles 2-5), we compute their observed success rate y_{human} for calls that were not quit at $t = 90$ and compare it to the algorithm's counterfactual success rate $y_{\text{algorithm}}(q)$ at the same quitting rate q. If $y_{\text{human}} < y_{\text{algorithm}}(q)$, it implies that the salesperson could have achieved higher success rate by adopting the algorithm's ranking. In other words, the salesperson is misranking calls by their failure risk.

For this comparison to be valid, we must assume that all salespeople draw from the same distribution of calls. In our setting, this assumption is plausible: prospect lists are randomly

Figure 15: Testing for salespeople misranking calls via contraction [Kleinberg et al., 2018]



Note: **The algorithmic decision rule curve is constructed by simulating quitting calls in the test set made by salespeople with quitting rates in the first quintile (analogous to lenient judges in [Kleinberg et al., 2018]) after $t = 90$ seconds, in decreasing order of calls' failure probability predicted by a fine-tuned GPT-4.1 model given the call transcript at $t = 90$. Each point is a non-lenient salesperson with a quitting rate in quintiles 2 to 5.**

assigned to salespeople, and we find no evidence of systematic differences in the distribution of predicted success probabilities of calls across salespeople.

Figure 15 overlays the observed performance of non-lenient salespeople (in the test set) as points on the algorithmic decision rule curve. We find that 48 of 60 non-lenient salespeople (80%) lie below the algorithmic decision rule curve (i.e., with $y_{\text{human}} < y_{\text{algorithm}}(q)$), indicating that their ranking of calls is outperformed by the algorithm's in terms of the success rate. This supports our earlier finding that salespeople mispredict call outcomes and highlights the value of decision support tools that incorporate predictive signals more systematically.

In sum, the contraction test provides independent validation that many salespeople misrank calls by their failure risk, even when given more time to gather information. This evidence strengthens our main claim: suboptimal quitting behavior reflects bounded rationality in risk assessment—reinforcing the need for decision support tools like stopping agents.