

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

October 10, 2025

Abstract

Salespeople frequently face dynamic screening decisions — whether to persist or quit the conversation with a prospect and move on to the next lead — yet little is known about how these decisions are made, whether they are efficient, or how to improve them. We study this problem in the context of high-volume outbound call centers, where time is scarce and failure is common. We formalize dynamic screening 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 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 language models and proprietary language model APIs. 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 why salespeople fall short, we find evidence of them overweighting salient rejection cues and mispredicting call failure risk, suggesting cognitive bounds on salespeople’s 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, conversational 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 participants at the New Data for Consumer Insights Conference (and especially our discussant Sanjog Misra), the Customer Intelligence Lab at Harvard, and the 2025 ISMS Marketing Science conference for their feedback, and an anonymous firm for providing the data. 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 outbound call centers — a \$97 billion global market [Grand View Research, Inc., 2024] — where every minute spent on a low-potential prospect could have been used to pursue a more promising lead.

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 call that is unlikely to succeed, 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, we find that the median length of a call that failed to end in a sale is almost 2 minutes. But just 1 minute into a call, a fine-tuned large language model can predict the risk of eventual failure to sell quite accurately (AUC=94%; details in Section 5). Quitting predictably risky calls 1 minute in could have saved salespeople significant time without sacrificing sales.

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]. Hence, 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 decides¹ whether and when to quit the conversation in real-time. 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.

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

By parameterizing the stopping policy with a generative large language model, 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 lack of scalability [Engstrom et al., 2020; Ahmadian et al., 2024].

We address this challenge by formulating policy estimation as an imitation learning problem. We show that optimal state-action trajectories can be inferred from potentially-suboptimal historical sales conversations. We then fine-tune a large language model to mimic these inferred optimal actions by minimizing the log loss of generating them given the transcript. In contrast with reinforcement learning approaches, our imitation learning approach is robust, scales to large language models, and is compatible with proprietary models only accessible via 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 an energy cross-selling campaign at our partner telecommunications firm. When permitted to quit 60 or 90 seconds into the 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 allowed to quit as early as 30 seconds into the call, increases expected sales by 37% by reducing the total call time by 54%. Model training and inference costs totaled about \$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’s decisions are highly predictable with simple machine learning models; a shallow random forest achieves a decision prediction AUC of 82% (but a call outcome prediction AUC of just 63%), indicating that salespeople’s decisions are potentially overly simple functions of phrases in the call transcript. Yet, the phrase most predictive of quitting by far — “*no me interesa*” (“I’m not interested”) — rarely appears early in the call, suggesting that salespeople needlessly delay quitting by waiting for this salient expression of disinterest.

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 model predicts selective shortening of 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 early in the call. Our analysis suggests that simply encouraging salespeople to shorten calls may be insufficient to improve their quitting decisions, and reinforces the need for algorithmic decision support.

Our research makes three key contributions, spanning substantive, methodological, and managerial dimensions. Substantively, we introduce and formalize the dynamic qualification problem and propose a sequential decision-making solution 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 significant expected sales gains and outperforms several natural benchmarks.

Methodologically, we develop a stopping agent that functions as a language model-based policy for optimal stopping with high-dimensional textual state representations. In doing so, we extend the natural language processing literature on detecting adverse conversational outcomes [Zhang et al., 2018] to the problem of 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 that estimate stopping policies using policy gradients [Venkata and Bhattacharyya, 2023], our approach scales to LLMs.

Managerially, we offer a practical and cost-effective solution for improving the effectiveness of high-volume outbound sales calls. 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 accessible only via APIs. Moreover, our diagnostic analysis suggests that salespeople struggle to predict call outcomes in real-time, underscoring the need for algorithmic decision support tools to alleviate their cognitive constraints.

The rest of the paper is organized as follows. Section 2 reviews related literature. Section 3 presents our generative language agent for optimal stopping. Section 4 applies our agent 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 efforts 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 large language models 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 leveraging large language models as collaborators in augmented intelligence systems [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 states, but implement their method with small recurrent neural network policies and non-textual states (we compare with their method in Section 4).

A broader stream of research applies reinforcement learning to align large language models with human preferences or maximize other reward functions, 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 conversational dynamics in our setting are externally governed by the customer–salesperson interaction.

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. The literature on consumer search also 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 large language models generate actions in addition to or instead of 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 language models to take actions 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 in 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 quitting decisions, we develop an algorithmic agent that observes the live conversation transcript and decides whether and when to quit. We build a *language agent*: a language model that generates decisions that optimize a long-term managerial objective. Unlike traditional conversational agents, our language agent does not interact with the prospect, but instead observes the dialogue between the salesperson and the prospect and quits the call 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 second 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 engaging a different prospect. This trade-off motivates formulating the task as an *optimal stopping problem* with textual states. We adapt the discrete-time finite-horizon optimal stopping formulation [Shiryaev, 1978] to our setting.

Formally, at each period $t \in \{1, \dots, T\}$, the state $s_t \in \mathcal{S}$ consists of the verbatim transcript of the conversation up to that period. A policy $\pi_\theta(a_t|s_t)$, parameterized by θ , observes s_t and selects an action $a_t \in \mathcal{A} = \{\text{wait}, \text{quit}\}$ 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 $a_T = \text{quit}$ 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 $\tau \in \{1, \dots, T\}$ denote the stopping time induced by a policy π_θ (which may vary per call), defined as the earliest time at which the action is quit:

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

²Importantly, *wait* implies the policy *doing nothing*; the conversation proceeds fully unaffected, as if the stopping agent did not exist. Formally, $\mathcal{P}(s_{t+1}|s_t, a_t = \text{wait}) = \mathcal{P}(s_{t+1}|s_t)$, and $\mathcal{P}(s_{t+1} = \text{end}|s_t, a_t = \text{quit}) = 1$. This distinguishes optimal stopping problems from general sequential decision-making problems, where actions may affect the next state. Our key innovation is to leverage this distinction for efficient policy estimation (Section 3.3). However, the tradeoff of adopting the canonical optimal stopping formulation is that our stopping agent cannot enforce or advise continuation if the salesperson decides to quit. Despite this tradeoff, we demonstrate substantial efficiency gains in Section 4.

The objective of the optimal stopping problem is to find an optimal stopping policy π_θ that maximizes the expected cumulative reward $J(\theta)$, defined as:

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

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, the firm may set w_t based on the opportunity cost of unit time (which can be a non-linear function of t), and q_τ to either zero (if the stopping agent quits before a sale was made) or to the profits a successful sale generates for the firm (if the stopping agent does not quit before a sale was made).

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 $V_t^*(s) = \max_\theta \mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^{\tau-1} w_t + q_\tau \mid s_1 = s \right]$ satisfy Bellman’s optimality conditions and can be computed recursively via backward induction:

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

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

³Based on Equation 2, one may consider a model-based policy that estimates $\mathbb{E}[V_{t+1}^*(s_{t+1}) | s_t, a_t = \text{wait}]$. However, this approach requires an accurate model of $\mathcal{P}(s_{t+1}|s_t)$, which is challenging to estimate in most real-world settings with finite data. Hence, modern policy estimation approaches are model-free and estimate the policy directly.

$|\mathcal{S}| \approx |\mathcal{V}|^L$. This exponential growth renders dynamic programming computationally infeasible 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 canonical remedy is to estimate a parameterized policy $\pi_\theta(\cdot)$. We therefore parameterize our stopping policy with a pretrained generative large language model to leverage the state-of-the-art natural language understanding ability of such models [Kaplan et al., 2020]. Estimating policies parameterized by large language models 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 large language models introduces well-known difficulties⁴ [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* (or behavioral cloning) [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 $\mathcal{D} = \{(s_1^i, a_1^i), \dots, (s_t^i, a_t^i)\}_{i=1}^N$ consisting of state-action pairs from an “expert” policy that maximizes the expected cumulative reward in

⁴Recent methods to align large language models with human preferences — including proximal policy optimization (PPO) [Schulman et al., 2017], group relative policy optimization (GRPO) [Shao et al., 2024], and other approaches for reinforcement fine-tuning [Lambert et al., 2024] — assume that the next state is formed by appending the policy’s action (i.e., the generated token) to the previous state. In our setting, however, this assumption does not hold: the next state is determined by the environment (i.e., the ongoing conversation between the customer and the salesperson).

Equation (1), imitation learning seeks a policy $\pi_{\hat{\theta}}(a|s)$ that minimizes the expected action mismatch:

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

that is, imitation learning finds a policy that replicates the behavior of this “expert” policy and, in doing so, maximizes the expected cumulative reward⁵.

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. In settings where such state-action trajectories are unavailable, this might appear prohibitive.⁶ Our key insight is that \mathcal{D} can be constructed using historical conversations. Specifically, we infer the cumulative reward-maximizing action a_t for each transcript prefix s_t by calculating the quitting and waiting rewards given the conversation outcome. We then train a large language model 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 a large language model.

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 large language models (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. In particular, it is readily compatible with *proprietary* language models that disallow custom loss functions and are accessible only via limited APIs.

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. We construct the dataset \mathcal{D} of optimal state-action trajectories using historical conversations. Specifically, we calculate, for each historical conversation, the quitting time τ that would have maximized the expected cumulative reward. This induces a state-action trajectory comprised of conversational states and their corresponding optimal actions, thereby representing an expert policy without needing access to expert decisions from salespeople.

⁵Kumar et al. [2022] prove that, given optimal state-action trajectories, reinforcement learning is no better than imitation learning in the worst case (i.e., their error bounds are identical). Foster et al. [2024] derive additional desirable properties of imitation learning when the action mismatch is quantified by the log loss.

⁶We do not assume that salespeople quit optimally and, in fact, provide evidence of their suboptimality in Section 5.

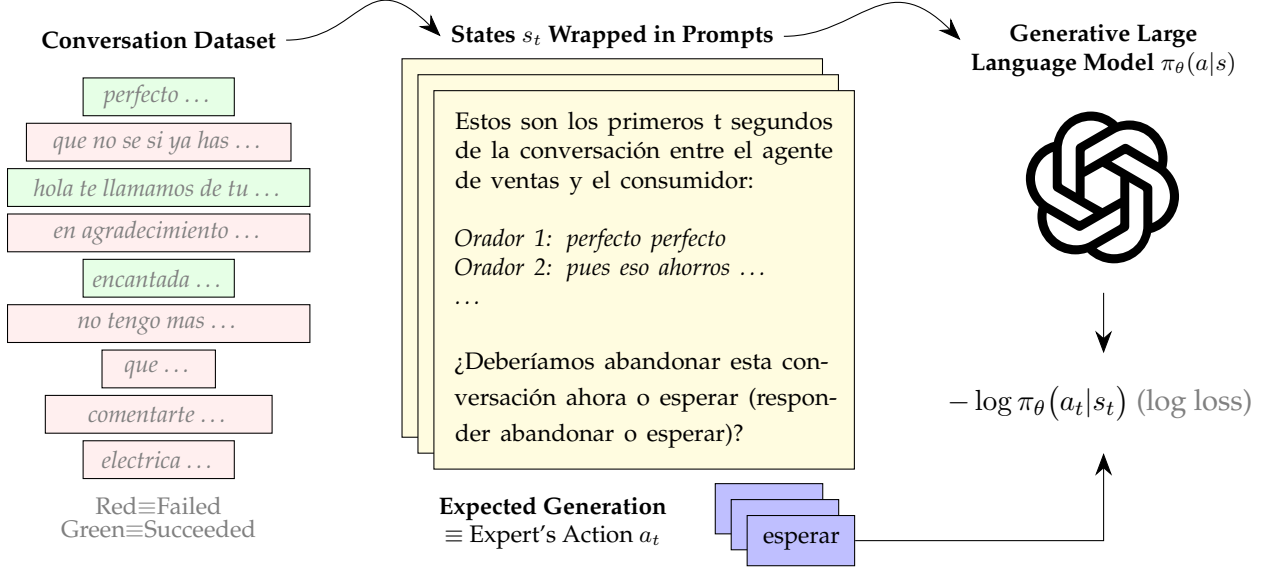


Figure 1: We transform conversations \mathcal{C} into “expert” demonstrations \mathcal{D} (Section 3.3.1), and train an imitation learning policy by fine-tuning a large language model π_θ to generate the “expert’s” action a_t for each state s_t wrapped in a prompt (Section 3.3.2). We threshold $\pi_\theta(a|s)$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

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 $a_t = \text{quit}$, or the conversation transcript until $t + 1$ if $a_t = \text{wait}$ (since waiting implies doing nothing). This structure distinguishes optimal stopping problems from general sequential decision-making problems, and our key innovation is to leverage this structure for imitation learning.

Formally, let $\mathcal{C} = \{(s_1^j, s_2^j, \dots, s_T^j, y^j)\}_{j=1}^M$ 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 $\tau \in \{1, \dots, T\}$, we compute the cumulative reward that would have been obtained by quitting at $t = \tau$ as,

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

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 $a_t^j = \text{wait}$ for $t < \tau_j^*$ and $a_{\tau_j^*}^j = \text{quit}$.

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, wait) and (s_2^j, 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 will not know how to act in this state.

We address this risk by following the data augmentation recommendation of Pomerleau [1988] in the context of autonomous vehicle control (i.e., how 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, quit) for each $t = \tau^* + 1, \dots, T$ and for each conversation j having optimal stopping time τ^* .

3.3.2 Expert Mimicry as Conditional Language Generation. The dataset \mathcal{D} of expert demonstrations comprises textual states associated with actions that can be viewed as natural language tokens (quit and wait). Hence, we cast the problem of mimicking the expert as that of conditional language generation. Specifically, we train a large language model policy π_θ by minimizing the empirical log-loss of its state-conditioned token generation over \mathcal{D} :

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

Thus, we reduce the problem of maximizing the expected cumulative reward in Equation (1) to that of fine-tuning a language model to generate the expert’s action a_t given the state s_t .

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

While a grid search over $\lambda_1, \dots, \lambda_T$ 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 $\mathcal{D} = \{(s_1^i, \dots, s_T^i, y^i)\}_{i=1}^N$, fine-tuned stochastic policy $\pi_{\hat{\theta}}$, horizon T

- 1: $\lambda_T \leftarrow 0$ ▷ Always quit at T
- 2: **for all** $(s_1^i, \dots, s_T^i, y^i) \in \mathcal{D}$ **do**
- 3: $R_T^i \leftarrow q_T$
- 4: **end for**
- 5: **for** $t = T - 1, T - 2, \dots, 1$ **do**
- 6: **for all** $(s_1^i, \dots, s_T^i, y^i) \in \mathcal{D}$ **do**
- 7: $R_t^i \leftarrow q_t$ ▷ Reward for quitting at t
- 8: $\tau \leftarrow \min\{u > t : \pi_{\hat{\theta}}(\text{quit} \mid s_u^i) \geq \lambda_u\}$ ▷ Next quitting time if agent waits at t
- 9: $R_\tau^i \leftarrow \text{reward of quitting at } \tau$ ▷ Known by induction for $\tau \in \{t + 1, \dots, T\}$
- 10: **end for**
- 11: $\lambda_t \leftarrow \arg \max_{\lambda \in [0,1]} \frac{1}{N} \sum_{i=1}^N (\mathbb{I}[\pi_{\hat{\theta}}(\text{quit} \mid s_t^i) \geq \lambda] R_t^i + \mathbb{I}[\pi_{\hat{\theta}}(\text{quit} \mid s_t^i) < \lambda] R_\tau^i)$
- 12: **end for**
- 13: **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. To facilitate adoption and further research, we provide an open-source framework to build, train, and evaluate stopping agents at stoppingagents.com. Having introduced the design and training of our stopping agent, we now apply it to real-world sales data to evaluate its performance.

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, with sales initiated through live phone calls by commissioned⁷ salespeople. These 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.⁸ For each call, we observe the complete 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	0.169	0.066	0.044
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 total sales performance.

⁷Their compensation consists of a base pay plus a bonus, across a two-shift workday with a set total of hours.

⁸We exclude 1,311 calls from the full sample that were missing transcripts.

4.2 Training our Stopping Agent

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

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 distribution⁹ (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 the steps described in Section 3.3, we train our stopping agent by fine-tuning the OpenAI GPT-4.1 model¹⁰ [OpenAI, 2025] on state-action pairs from optimal state-action trajectories inferred from calls in the training set. Each conversation state (i.e., partial call transcript) is wrapped in a prompt¹¹, as illustrated in Figure 1. We tune the action thresholds on the validation set, and report all performance metrics on the held-out test set.

4.3 Evaluating our Stopping Agent’s Performance

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

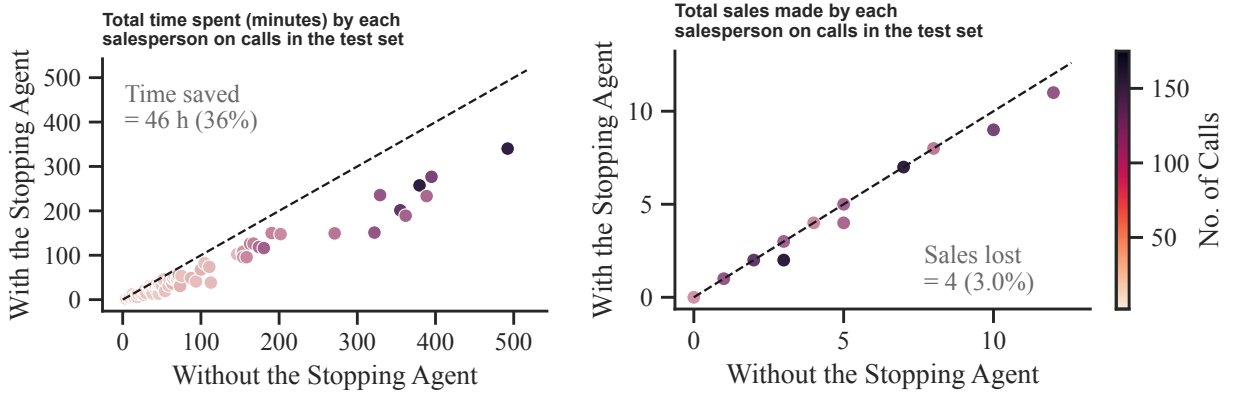
⁹This is a reasonable assumption in high-volume sales settings, such as in telemarketing, where prospect supply is ample. In other settings, our stopping agents’ reward structure can readily be configured to include factors such as labor cost reductions, improved salesforce morale from greater efficiency, or (as noted by managers at our partner firm) improved customer satisfaction from salespeople spending less-time on low-potential calls.

¹⁰We 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.

¹¹We found that performance is insensitive to the exact prompt, since fine-tuning tends to eliminate its effect as a prior.

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 2. Each point represents a salesperson, with color intensity indicating their call volume in the test set. Points on the $x = y$ diagonal correspond to no change in time or sales from using the stopping agent. For instance, in the left panel, points below the diagonal are salespeople who would have saved time using our stopping agent.

Figure 2: Total time spent (left) and total number of sales made (right) by each salesperson, both with (y -axis) and without (x -axis) our stopping agent with $T = 2$ 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.

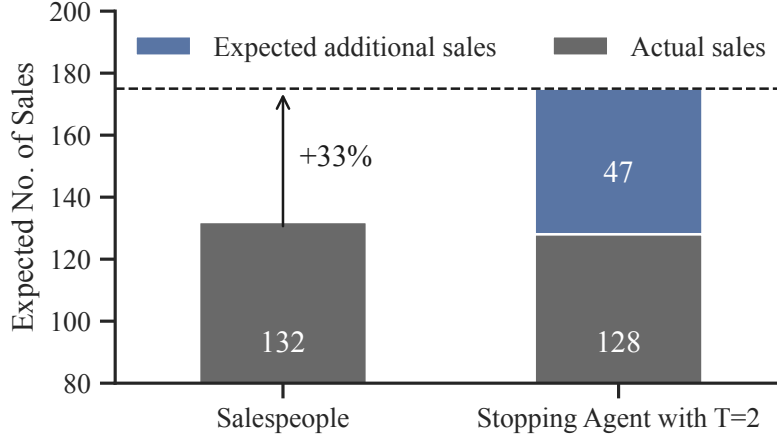
Salespeople spent a total of 128 hours on calls in the test set. Our stopping agent reduces call time by 1.37% to 71.6% per salesperson, totaling 46 hours across all salespeople (a 36% reduction in total call time). This reduction comes at the cost of 4 sales, or 3% of the 132 sales in the test set.

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:

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

As in Section 4.2 (footnote 9), 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). In practice, this assumption is reasonable when the pool of eligible leads exceeds the number of leads actually contacted, as is the case in our empirical setting.

Figure 3: 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).

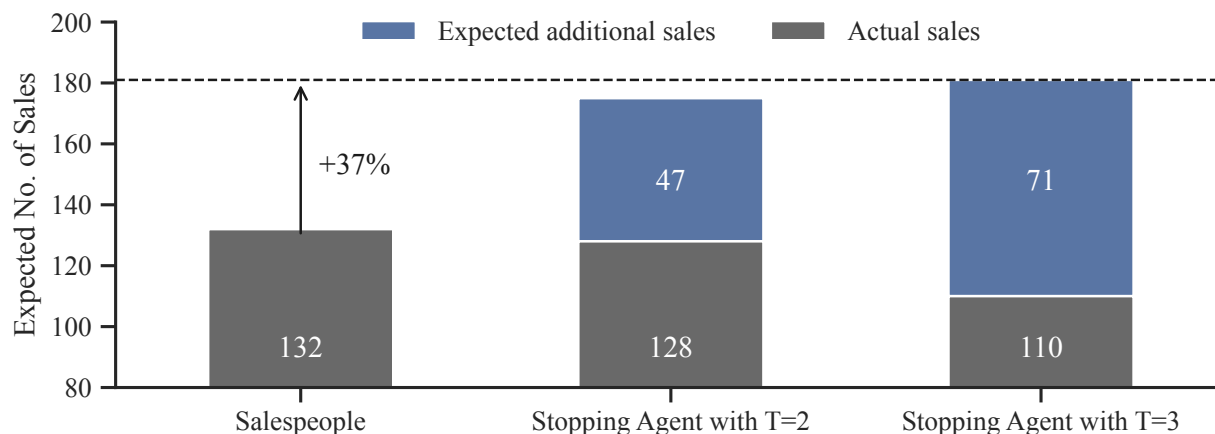
We visualize the actual and expected number of sales on the test set calls for salespeople and for our stopping agent in Figure 3. With only $T = 2$ decision opportunities at $t = 60$ and $t = 90$ seconds, our stopping agent delivers a 33% increase in expected sales (from 132 to 175 sales). Of these, 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.

Extending the Decision Horizon This strong performance naturally raises the question of whether quitting earlier could unlock even greater value. To explore this possibility, we introduce a third decision point at $t = 30$, retrain our stopping agent, and reevaluate its performance. As shown in Figure 4, 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 any number of decision opportunities¹². We next explore whether the benefits of our stopping agent vary by salesperson effectiveness, and benchmark its performance against alternative stopping policies.

¹²In 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 4: 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 in the test set ($\approx 4\%$). 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
Expected Sales Gain (%)	—	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 test set 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 the salesforce.

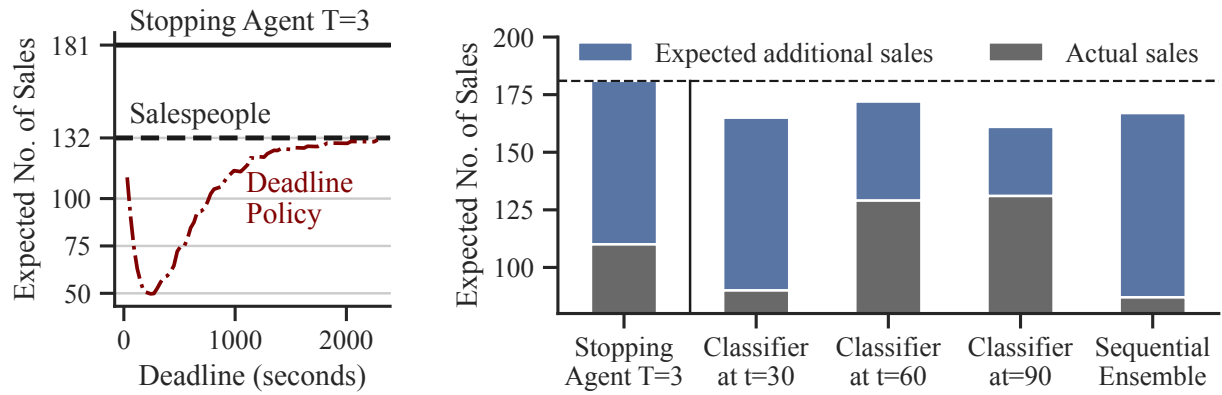
4.4 Evaluating Alternative Stopping Policies

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 baseline 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 5a 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 call content-agnostic deadline policy underperforms both salespeople and our stopping agent. This suggests that a simple rule advising agents to quit if a call lasts more than t seconds is unlikely to improve sales effectiveness.

Figure 5: Expected number of sales by our stopping agent and several myopic 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. 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 5b, although the classifier-based

policies outperform salespeople, they perform worse than our stopping agent despite being built on the same GPT-4.1 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 5b, this ensemble also underperforms our stopping agent.

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

4.4.2 A State-of-the-Art Reinforcement Learning Method. We benchmark our stopping agent against the state-of-the-art deep reinforcement learning approach for optimal stopping problems: the optimal stopping policy gradients (OSPG) algorithm [Venkata and Bhattacharyya, 2023].

OSPG is designed for recurrent neural network (RNN) policies, and not large language model 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 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)	Expected Additional Sales	Expected Sales Gain (%)
Policy size = 3092×1 , ~ 10 M parameters	117	109	20	4%
Policy size = 3092×5 , ~ 50 M parameters	95	99	29	-6%
Policy size = 3092×10 , ~ 100 M parameters	109	117	11	-9%

Note: The policy size $H \times D$ indicates H hidden units in each of D hidden layers. Following 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 Adam optimizer with early stopping on the validation set.

The OSPG-trained policies underperform our stopping agent. Notably, larger policy sizes (>100 M parameters) degrade performance, with expected sales losses exceeding 10%. We also find

that many training runs fail due to the reward plateauing or collapsing to zero, consistent with the well-known 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

Open-source language models can be hosted by firms on-premise, and offer a privacy-preserving alternative to language models hosted remotely via APIs. However, open-source language models tend to be less performant than proprietary language models. We now evaluate whether our stopping agent can provide efficiency gains with such 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)	Expected Additional Sales	Expected Sales Gain (%)
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 delivers expected sales gains between 10% and 16%. These results showcase that even under privacy and computational constraints, firms can use our stopping agent to improve sales efficiency. They also demonstrate the versatility of our stopping agent in leveraging both proprietary and open-source language models.

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 *6 months* after the original campaign. In this *out-of-sample* campaign (since we did not use any calls from this campaign to train our stopping agent), the firm targeted subscribers of a lower-cost sub-brand (similar to how Mint Mobile relates to T-Mobile).

The out-of-sample campaign comprises 8,334 first-contact calls made by 153 salespeople over 1 month. Only 31 (i.e., 20%) of these salespeople were also part of our original campaign. While salespeople spent about the same time per call in the original and out-of-sample campaigns (195 vs. 196 seconds), their success rate was lower in the out-of-sample campaign than in the original campaign (4.9% vs. 5.5%).

The results in Table 5 show that our stopping agent continues to improve sales efficiency in this out-of-sample campaign without retraining. Our stopping agent achieves a 10% ($T = 2$) and 16% ($T = 3$) increase in expected sales relative to salespeople (corresponding to 39 and 63 additional sales, respectively), despite the temporal gap and the differences in the target subscriber base between the original and out-of-sample campaigns.¹³

Table 5: Evaluating our stopping agent on an out-of-sample campaign.

	No. of Sales	Total Time (h)	Expected Additional Sales	Expected Sales Gain (%)
Salespeople	405	454	—	—
$T = 2$ Stopping Agent	328	323	39	10%
$T = 3$ Stopping Agent	275	237	63	16%

These findings suggest that the conversational patterns and decision thresholds learned by the stopping agent remain robust under distribution shift. In practice, this robustness reduces the need for frequent retraining every time the campaign, salesforce composition, and target customer base changes, thus enhancing the practicality of using our stopping agent in the field.

¹³If we adjust the decision thresholds to use the average call duration and success rate in the out-of-sample campaign, but keep the trained policy from the original campaign, the expected sales gain produced by our stopping agent further increases to 11% ($T = 2$) and 17% ($T = 3$).

5 Diagnosing Salespeople’s Quitting Decisions

Having demonstrated that our stopping agent delivers substantial and durable performance improvements, we now turn to diagnosing why salespeople’s quitting decisions fall short. Since identifying salespeople’s behavioral primitives directly is challenging without experimental data, we approach this question by empirically examining the informational and behavioral patterns associated with salespeople’s quitting decisions.

Specifically, in Section 5.1, we train interpretable machine learning predictors of salespeople’s quitting decisions and find evidence suggesting that salespeople rely on simple decision rules that overweight a few salient phrases that occur late in the call. In Section 5.2, we analyze how salespeople adjust which calls they quit under heightened time pressure, and find evidence consistent with salespeople being insensitive to call failure risk.

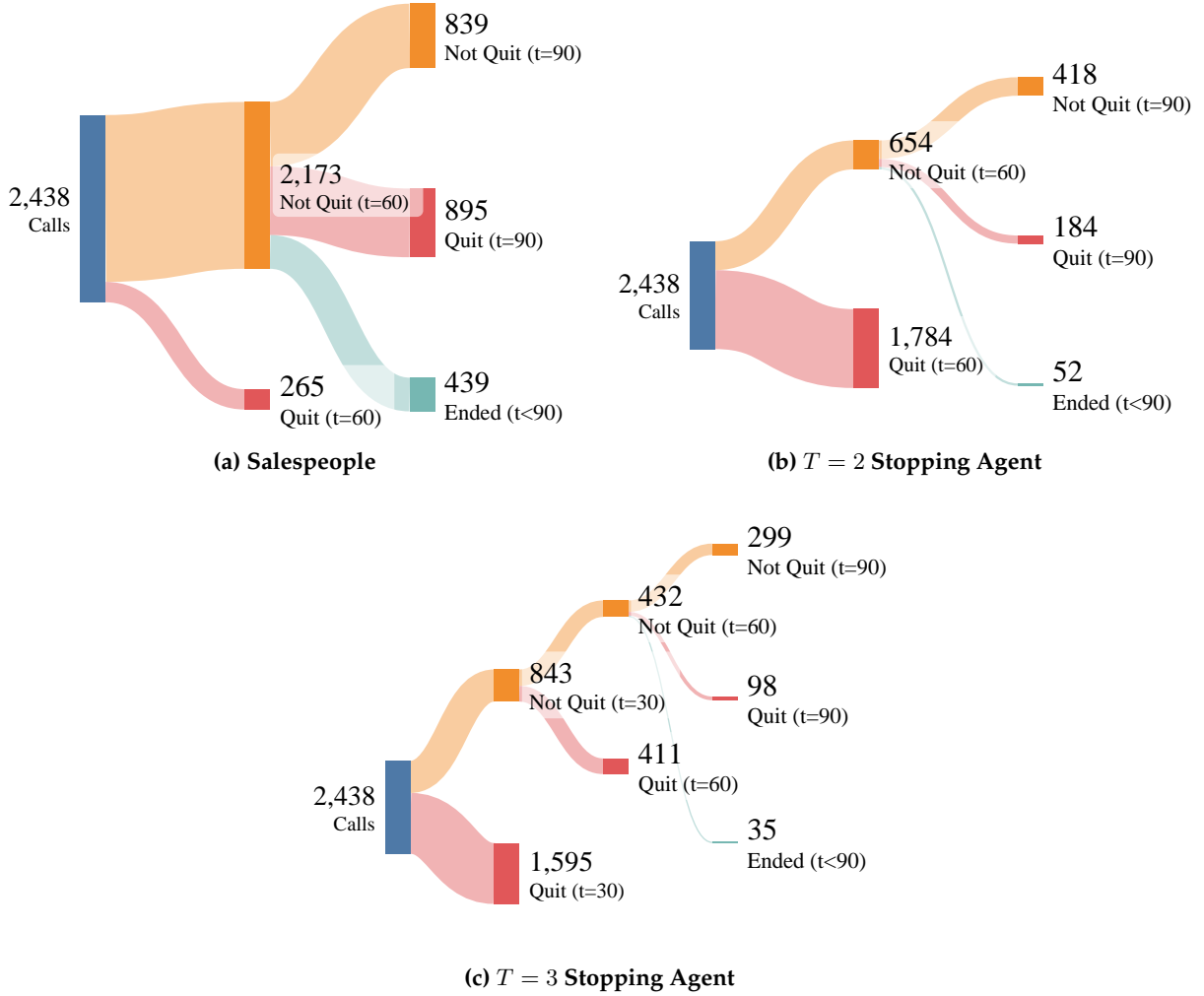
Together, our results suggest cognitive bounds on salespeople’s ability to disqualify consumers early in the call while also having to engage with the consumer, handle the consumer’s objections, and perform other selling tasks. This motivates the need for algorithmic support to lower the cognitive burden of multiple selling tasks and improve real-time disqualification decisions.

5.1 Probing When and Why Salespeople Quit with Interpretable Machine Learning

5.1.1 Examining when salespeople quit. We first visually examine when salespeople quit for call in the test set. To facilitate comparison with our stopping agent, we encode the salesperson’s quitting decision at $t \in \{30, 60, 90\}$ as a binary indicator of whether the call was longer than $t + \Delta$ seconds (we set $\Delta = 10$ based on the average duration of closing salutations, but our results are robust to other values of Δ).

In Figure 6a, we visualize salespeople’s quitting decisions as a Sankey diagram. Figure 6a shows that salespeople rarely quit before 60 seconds into a call: they quit *no* calls at $t = 30$, and just 11% of calls at $t = 60$. At $t = 90$, salespeople quit 52% of the surviving calls. Given their superior performance, we compare with our stopping agents as a benchmark. Our $T = 2$ stopping agent (visualized in Figure 6b) quits 75% of the calls at $t = 60$, and 31% of the surviving calls at $t = 90$. Our $T = 3$ stopping agent (visualized in Figure 6c) also tends to quit early. This is in contrast with salespeople, who appear to delay quitting until later in the call.

Figure 6: Sankey diagram of quitting decisions by salespeople and by our stopping agents.



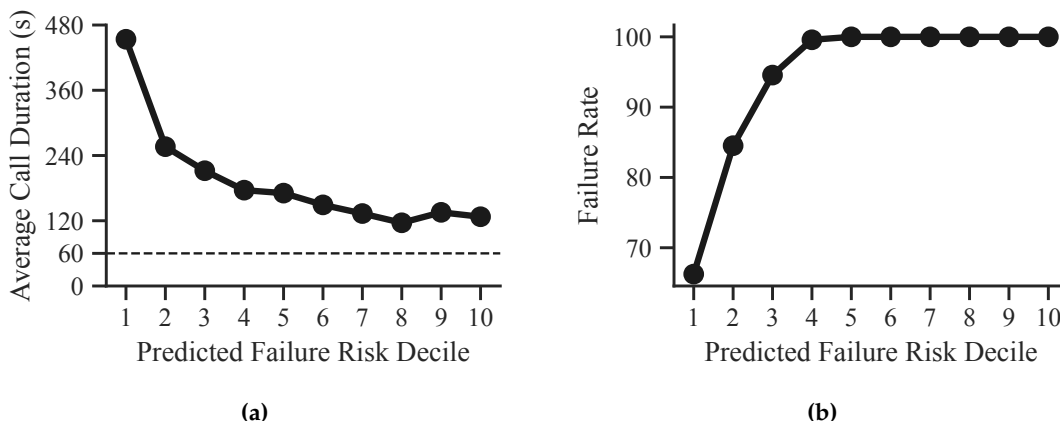
Note: Salespeople did not quit any calls at $t = 30$, so we omit that decision opportunity from the Sankey diagram. 439 calls in the test set that were not quit by salespeople at $t = 60$ (by our definition) ended before $t = 90$.

5.1.2 Exploring why salespeople delay quitting. Figure 6 suggests that, relative to our stopping agents, salespeople hesitate to quit. However, the figure does not reveal *why* salespeople hesitate.

Does a lack of information explain hesitation? A plausible reason for salespeople’s hesitation is the lack of information available early in the call to accurately assess the expected value of quitting versus continuing the call. Although the superior performance of our stopping agents already contradicts this explanation, we provide additional evidence by building a predictor of call failure risk. Specifically, we fine-tune (on the training set) GPT-4.1 to predict eventual call failure given the

first 60 seconds of the call text, sort the test set calls into deciles of predicted failure risk, and plot the average time spent by salespeople on the calls in each decile. Our results are in Figure 7a.

Figure 7: Examining how salespeople react to early linguistic signals of eventual call failure.



Note: Failure risk of calls in the test set predicted by a GPT-4.1 classifier fine-tuned on calls in the training set.

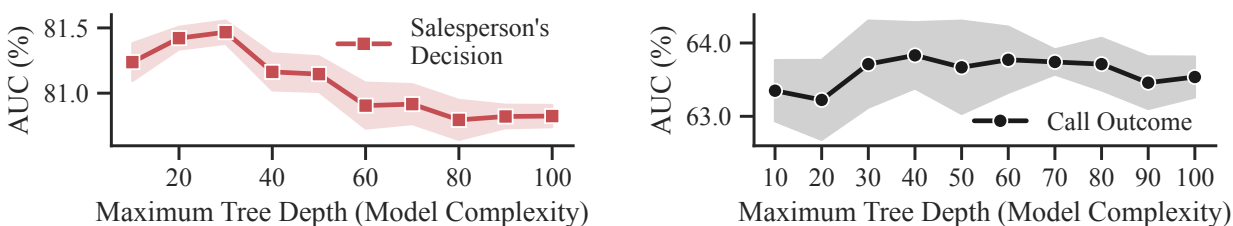
Figure 7a shows that salespeople spend less time on calls that are predictably risky, suggesting that they indeed react to early linguistic signals of eventual call failure. Yet, they seem to *under-react*. On calls in the six highest predicted failure risk deciles, salespeople spent about 55 hours and two minutes per call on average. However, as Figure 7b shows, *all* of these calls ultimately failed. These predictably-risky calls could have been abandoned just 60 seconds in without losing any sales.

What predicts salespeople quitting? To further explore the behavioral model underlying salespeople’s quitting decisions, we train interpretable machine learning models to predict whether a salesperson will quit at $t = 60$ based on the call transcript text up to that point. Specifically, we train random forest models of different complexities (by varying their maximum tree depth) and use the 10,000 most frequent unigrams and bigrams as the input features for the random forests.

Figure 8 (left) shows the held-out AUC (a measure of predictive performance, where 100% is best) of predicting quitting decisions with increasingly complex models. To put the performance of predicting quitting decisions into perspective, we also show the held-out AUC of predicting call outcomes in Figure 8 (right).

We highlight two key insights from Figure 8. First, salespeople’s quitting decisions are significantly more predictable than call outcomes: the held-out AUC of predicting quitting decisions is

Figure 8: Predicting salespeople quitting at $t = 60$ (left) and eventual call outcomes (right) given the first 60 seconds of the call transcript using random forest models of different 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).

higher than that of predicting call outcomes by over 17 percentage points. Second, decision prediction performance deteriorates sharply with increasing model complexity, whereas call outcome prediction performance remains relatively stable with increasing model complexity.

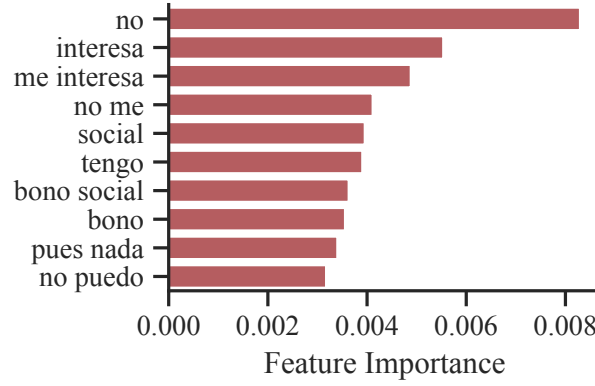
These findings suggest that salespeople’s quitting decisions follow simple rules, which are well captured by shallow random forests but overfit by deep ones. In contrast, call outcomes appear to follow more complex rules that are not captured as well even by deep random forests. The simple rules governing salespeople’s quitting decisions may not reflect the complex rules that determine call outcomes. Together, these findings suggest that salespeople are cognitively constrained, which may explain why salespeople’s quitting decisions fall short relative to our stopping agent’s.

We leverage the interpretability of random forests to uncover the linguistic cues that potentially drive salespeople’s quitting decisions. Specifically, we extract the most predictive features from our best random forest model of salespeople’s quitting decisions by computing the Gini importance [Breiman et al., 2017] of each feature. Figure 9 lists the 10 most important features.

Notably, the most important features all come from the phrase “*no me interesa*” (i.e., “I’m not interested”), suggesting that salespeople tend to quit in response to this explicit expression of disinterest, and tend to continue the call otherwise. Further, the feature importance distribution is skewed towards the phrases comprising “*no me interesa*”, suggesting a higher reliance on this expression of disinterest than on other linguistic cues.

“I’m not interested” is a special phrase in sales — it is the canonical example of a sales objection and appears as such in nearly all sales training materials. Thus, “*No me interesa*” is potentially *salient*, and salience is a well-known source of decision-making bias [Tversky and Kahneman, 1974].

Figure 9: Most important features (by Gini importance [Breiman et al., 2017]) in the best random forest model predicting salespeople’s quitting decisions at $t = 60$.



If salespeople over-rely on “*no me interesa*” relative to other linguistic cues, they may quit too late, because “*no me interesa*” appears during the first 60 seconds of only 7.4% of the calls that eventually failed. We now further investigate whether salespeople overweight “*no me interesa*”.

5.1.3 Evidence of salespeople overweighting “*no me interesa*”. Waiting for “*no me interesa*” and ignoring other informative linguistic cues could produce quitting delays, given the rarity of “*no me interesa*” appearing in the first 60 seconds. We assess potential overweighting of “*no me interesa*” by comparing the correlation of the top 10,000 unigrams and bigrams with salespeople’s quitting decisions and the predicted call failure risk at $t = 60$ (as used in Figure 7a).

Figure 10 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 call failure risk ($r \in [0.03, 0.06]$), and are clear outliers relative to the other unigrams and bigrams. Hence, Figure 10 suggests that “*no me interesa*” has an outsized influence on salespeople’s quitting decisions relative to other phrases and to its association with predicted call failure risk.

What makes our stopping agent quit? Figures 6b and 6c show that our stopping agent tends to quit earlier than salespeople, and thus achieves superior performance. We now replicate the previous analysis of salespeople for our stopping agents to understand why they quit.

In Figure 11a, we show the top 10 most predictive features from our best random forest model of our $T = 2$ stopping agent’s quitting decisions at $t = 60$ based on their Gini importance (analogous

Figure 10: Correlation of each of the top 10,000 unigrams and bigrams (red point) with the call failure risk predicted by GPT-4.1 at $t = 60$ (y -axis) and with salespeople quitting at $t = 60$ (x -axis).



to Figure 9)¹⁴. The distribution of feature importances is less skewed than in Figure 9, and appears to span several linguistic indicators of a lack of conversational progress. For example, “no”, “pero” (“but”), “digame” (“tell me”), and “quien” (“who”) indicate conversations stalled at the consumer identifying who the caller is and what they want, instead of progressing to affirmations of interest or to curiosity about the product.

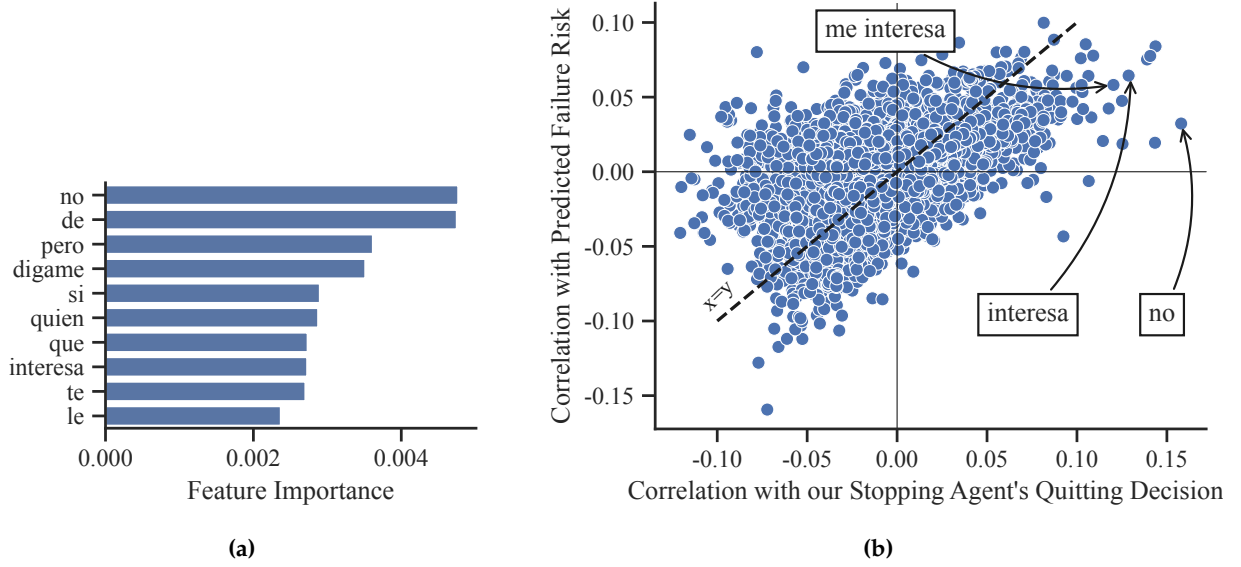
In Figure 11b, we replicate Figure 10 for our $T = 2$ stopping agent, and find that “no me interesa” does not appear to have an outsized association with quitting relative to other phrases and to its association with predicted call failure risk at $t = 60$. This further suggests that our stopping agent is less biased by the salient “no me interesa” phrase than salespeople.

To better interpret why our stopping agent quits, we estimate penalized¹⁵ logistic regression models 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 text before t . We report the unigrams and bigrams most positively and most negatively associated with

¹⁴Note 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.

¹⁵We employ L_1 and L_2 penalization of the estimated coefficients to alleviate potential overfitting due to the large number of correlated independent variables in the regression (i.e., 10,000 unigrams and bigrams).

Figure 11: (a) Most important features (by Gini importance [Breiman et al., 2017]) in the best random forest model predicting our $T = 2$ stopping agent’s quitting decisions at $t = 60$; (b) Correlation of each of the top 10,000 unigrams and bigrams (red point) with the call failure risk predicted by GPT-4.1 at $t = 60$ (y -axis) and with our $T = 2$ stopping agent quitting at $t = 60$ (x -axis).



the quitting decision in Table 6. In contrast with Figures 10 and 11b, Table 6 reports *conditional* correlations (i.e., given the other unigrams and bigrams in the call before t).

Table 6: Top words and their logistic regression coefficients by decision-maker and decision opportunity.

Stopping Agent						Salespeople			
$t = 30$		$t = 60$		$t = 90$		$t = 60$		$t = 90$	
digame	2.4	no	0.9	no	3.2	no	4.2	no	6.6×10^{-4}
quien	2.2	de	0.6	tengo	2.7	interesa	2.4	no no	3.0×10^{-4}
le	2.1	interesa	0.5	bien	2.5	lo	2.0	interesa	2.2×10^{-4}
no	2.0	no no	0.5	como	2.5	pero	1.8	tiene	2.0×10^{-4}
compania	1.8	me interesa	0.4	claro	2.2	me interesa	1.7	me interesa	1.9×10^{-4}
soy yo	-1.9	dias	-0.3	si	-2.5	linea	-1.2	la compa��a	-1.4×10^{-4}
si soy	-1.6	se si	-0.3	hay	-1.8	fibra	-1.1	los	-1.2×10^{-4}
me ha	-1.4	te	-0.2	asi	-1.7	moviles	-1.0	telefon��a	-1.2×10^{-4}
vale	-1.4	por aqui	-0.2	hace	-1.6	la fibra	-1.0	vale	-1.1×10^{-4}
si yo	-1.4	va	-0.2	sigue	-1.5	dias	-0.9	las	-1.0×10^{-4}

Note: Coefficients are from a penalized logistic regression model of the stopping agent’s (left) or salesperson’s (right) quitting decision at t on the top 10,000 unigrams and bigrams in the test call before t . The top panel lists the 5 words with the largest positive coefficients, and the bottom panel lists the 5 words 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.

Table 6 reveals several insights. Notably, salespeople’s quitting decisions at both $t = 60$ and $t = 90$ are overwhelmingly associated with unigrams and bigrams in the phrase “*no me interesa*” (“I’m not interested”), suggesting that salespeople continue to delay quitting across decision opportunities until they hear “*no me interesa*”. In contrast, our stopping agent progresses through distinct phases of identifying whether to quit.

At $t = 30$, perfunctory responses from the prospect 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 emerging cooperation and are negatively associated with our stopping agent quitting. Taken together, these patterns suggest that our stopping agent applies a screening rule at $t = 30$: unless the prospect provides affirmative engagement cues in the first 30 seconds, the stopping agent is likely to quit.

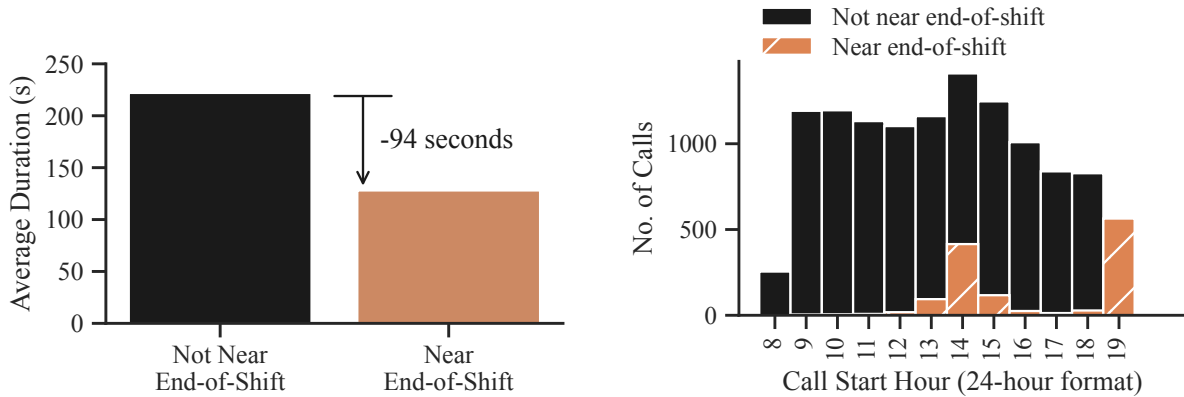
At $t = 60$, the phrases positively associated with our stopping agent quitting shift to explicit indicators of disinterest, such as “*no*” and “*interesa*” (“interested”), while the negative associations now come from phrases that typically precede volunteering or asking for more information, such as “*se si*” (“if”) and “*por aqui*” (“here”). Thus, it seems that between $t = 30$ and $t = 60$, our stopping agent moves from quitting based on the *absence* of engagement to the *presence* of disinterest.

At $t = 90$, only 397 calls survive for our stopping agent: 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 most negatively associated phrases all indicate conversational progress.

5.2 Assessing Quitting Behavior Under Different Opportunity Costs of Time

We next examine whether salespeople adjust their quitting behavior in response to variation in their opportunity cost 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 predicted risk of failure, and preserve time for those with a low predicted 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 call failure risk.

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.

Operationalizing variation in opportunity costs of time To measure 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 a higher opportunity costs due to salespeople eager to finish on time, avoid starting long conversations, or meet their daily quotas efficiently.

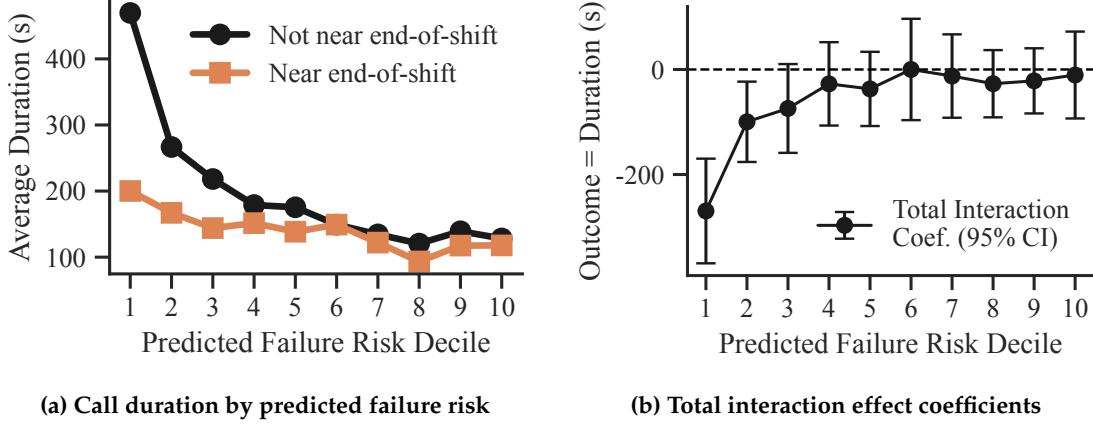
We compute the time between each call’s start and the end of the salesperson’s shift, and bin test set calls into deciles based on their time-till-end-of-shift. We denote calls in the first decile as “near end-of-shift”. Figure 12a shows that “near end-of-shift” calls are significantly shorter than earlier calls,¹⁶ suggesting that salespeople indeed face higher opportunity costs of time during those calls. Figure 12b shows the calls’ start times and confirms that “near end-of-shift calls” are made near the end (i.e., the last hour) of shifts (which end at 3 p.m. and 8 p.m.).

Do salespeople allocate call time in a risk-sensitive manner? We now ask whether salespeople respond to the higher opportunity cost of time by reallocating time more effectively across calls with different predicted failure risks. Figure 13a plots average call durations by predicted failure risk decile (in the test set), separately for near end-of-shift calls and for those made earlier in the shift. If salespeople were behaving in a risk-sensitive manner, we would expect them to shorten

¹⁶We confirm the significance of this difference with a regression of call duration on a dummy for “near end-of-shift” calls. The estimates show that “near end-of-shift” calls are 94 seconds shorter on average than other calls ($p < 0.001$).

high-risk calls more aggressively. Instead, we find that the call duration declines most sharply 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.



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

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

$$\text{Duration}_i = \alpha_0 + \gamma_0 s_i + \sum_{j=2}^{10} \alpha_j p_{ij} + \sum_{j=2}^{10} \beta_j p_{ij} s_i + \epsilon_i, \quad (5)$$

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 having started 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 risk¹⁷.

5.3 Summary and Discussion

Our findings suggest that salespeople rely heavily on simple functions of a few salient but weakly predictive cues (most notably, the phrase “no me interesa”) and do not adjust their quitting behavior in response to elevated time costs. When faced with higher opportunity costs 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 bounded rationality: 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, most calls do not result in a sale, yet they consume substantial time and resources. This inefficiency poses a core managerial challenge: how to identify, in real time, which conversations are worth continuing and which should be abandoned. We address this problem by introducing a stopping agent: a generative *language agent* 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 in both efficiency and effectiveness. 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.

¹⁷In 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.

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 often rely on salient but weakly predictive cues (e.g., “no me interesa”) and do not reallocate effort efficiently under time pressure. These patterns suggest that cognitive bounds are a key constraint in dynamic qualification decisions, and 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 conversion¹⁸. 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 the stopping agent. Given the versatility of the underlying language models, our approach could be extended to support a broader set of conversational decisions beyond quit and wait, for example, recommending scripted responses (à la Google’s smart-reply [Kannan et al., 2016]), asking clarifying questions, or suggesting pauses. Such multi-action policies would allow the agent to not only determine whether to continue, 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

¹⁸Discussions with our partner firm suggest that boosting short-term sales was indeed the goal of the campaign

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 salesforce 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 large language models 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. If, however, salespeople spend too little time on calls (i.e., by quitting too early), our stopping agent is unable to advise salespeople to persist. Addressing this limitation would require *offline-to-online* reinforcement learning, where offline imitation learning is followed by online policy improvement through direct interactions with salespeople [Yue et al., 2024].

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]. The extent to which such structure can be incorporated into our stopping agent remains an open question.

Summary In sum, we show that the decision to quit 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 sales 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 transparent, 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 the Cornell Atkinson Center for Sustainability and 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 <i>Diplomacy</i> 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.
- 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 Research, Inc. 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. *Retention Management using Dynamic Off-Policy Policy Learning (May 2, 2024)*, 2024.

- Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should I Run Offline Reinforcement Learning or Behavioral Cloning? In *International Conference on Learning Representations*. 2022.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 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. Alvin: 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, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/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 defendant 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.

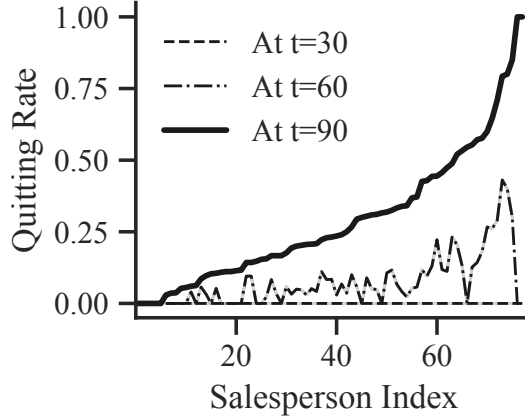
This test relies on the assumption that the decision does not itself affect the counterfactual outcome, only who is selected into the decision. For example, in Kleinberg et al. [2018], a judge releasing or detaining a defendant does not change the defendant’s underlying propensity to commit a crime; 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.

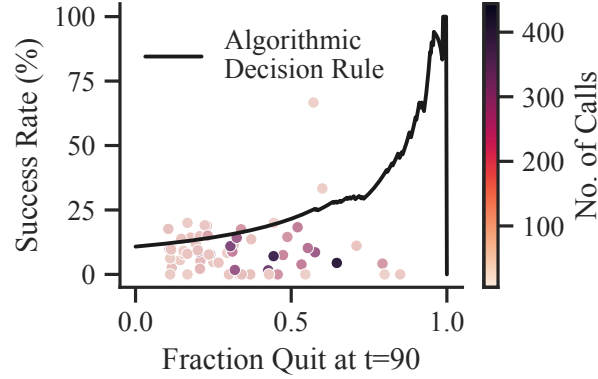


To construct the algorithmic decision rule, we simulate quitting calls from the test set made by the most lenient salespeople, those in the first quintile of the quitting rates distribution at $t = 90$, in decreasing order of predicted failure risk. Predictions are generated by a fine-tuned GPT-4.1 model 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.