Agentic AI for Frontline Decision-Making:

When Should Customer Conversations Stop?

Eva Ascarza
Harvard Business School

April 7, 2026

Learning When to Quit in Sales Conversations

Emaad Manzoor (Cornell)
Eva Ascarza (Harvard)
Oded Netzer (Columbia)

"But this ice is artisanal and glacier-sourced!”

High-Volume Outbound Call Centers

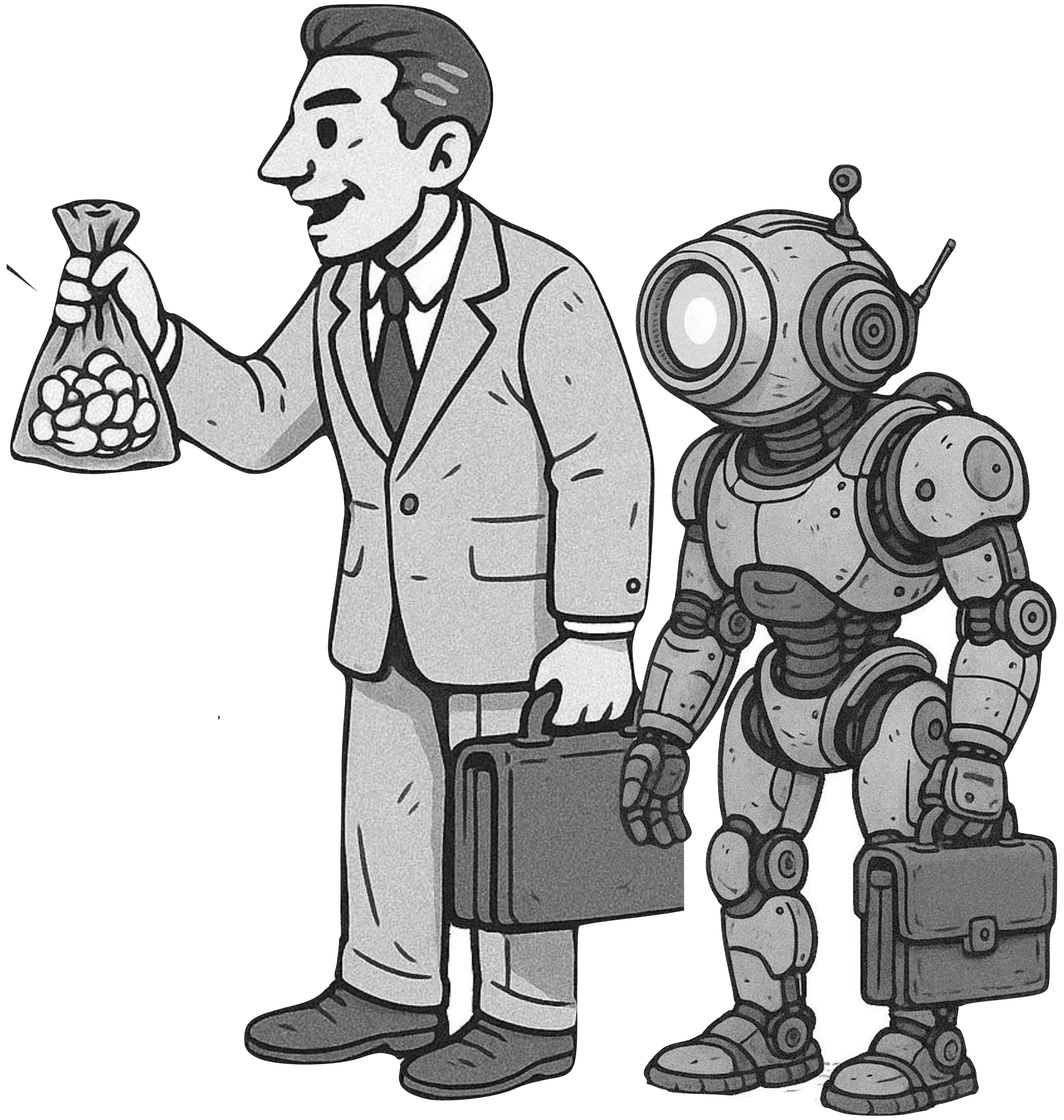
- $315B market globally (expected $500B in 2027)
- Low success rates: ~2% on average
- High Daily Call Volumes: 45 to 60 calls per day
- Short window to engage, between 5 to 10 minutes per prospect
- A lot of attention to how to sell

When to quit: Neglected sales decision

Sources: https://www.cognitivemarketresearch.com/call-centers-market-report | https://financesonline.com/cold-calling-statistics

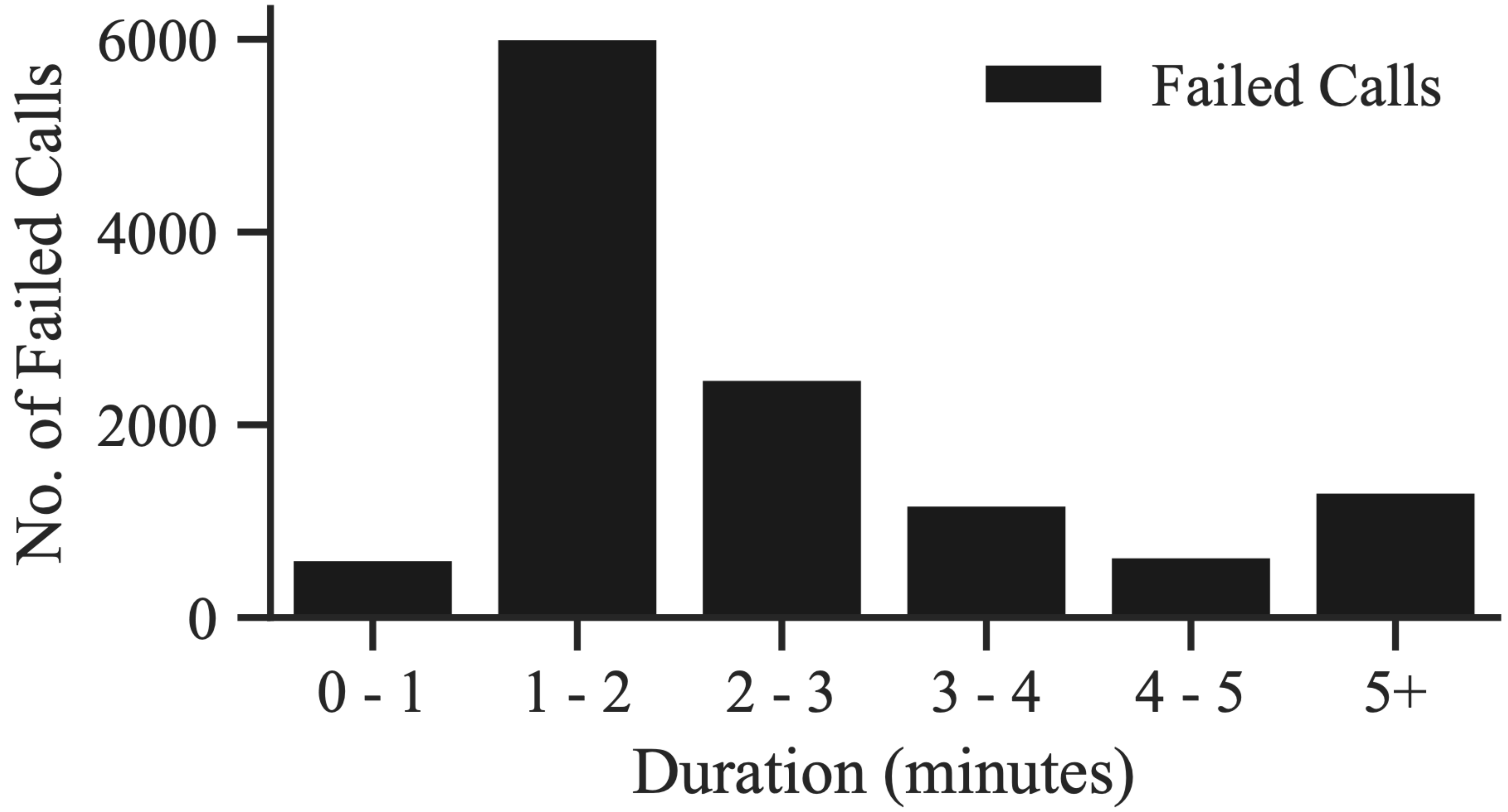
This Work: Algorithmic Disqualification

- We propose a stopping agent — a generative language agent that observes live conversations and decides whether and when to quit
- We evaluate our stopping agent on sales calls from a large European firm — +37% expected sales gain by reducing failed call time by 54%
- We find evidence of cognitively-constrained salespeople — motivates need for decision support



Data & Opportunity

- Dataset: 11,627 first-contact outbound calls by 79 salespeople at a European telecom company over 1 month
  - Cross-selling campaign to sell energy, success ≡ confirmed energy contract
- 94.5% of calls fail to end in a sale, average failed call is 3 minutes long
  - 23 days spent on failed calls in total



Salespeople spend significant time on failed calls

Do calls have early indicators of failure?

You are a sales agent predicting call success.

This is the first 60 seconds of the call between the salesperson Speaker 0 and consumer Speaker 1:

Speaker 0: ...

Speaker 1: ...

Will this call end in a sale? (answer yes or no):yes

Correct Response

2

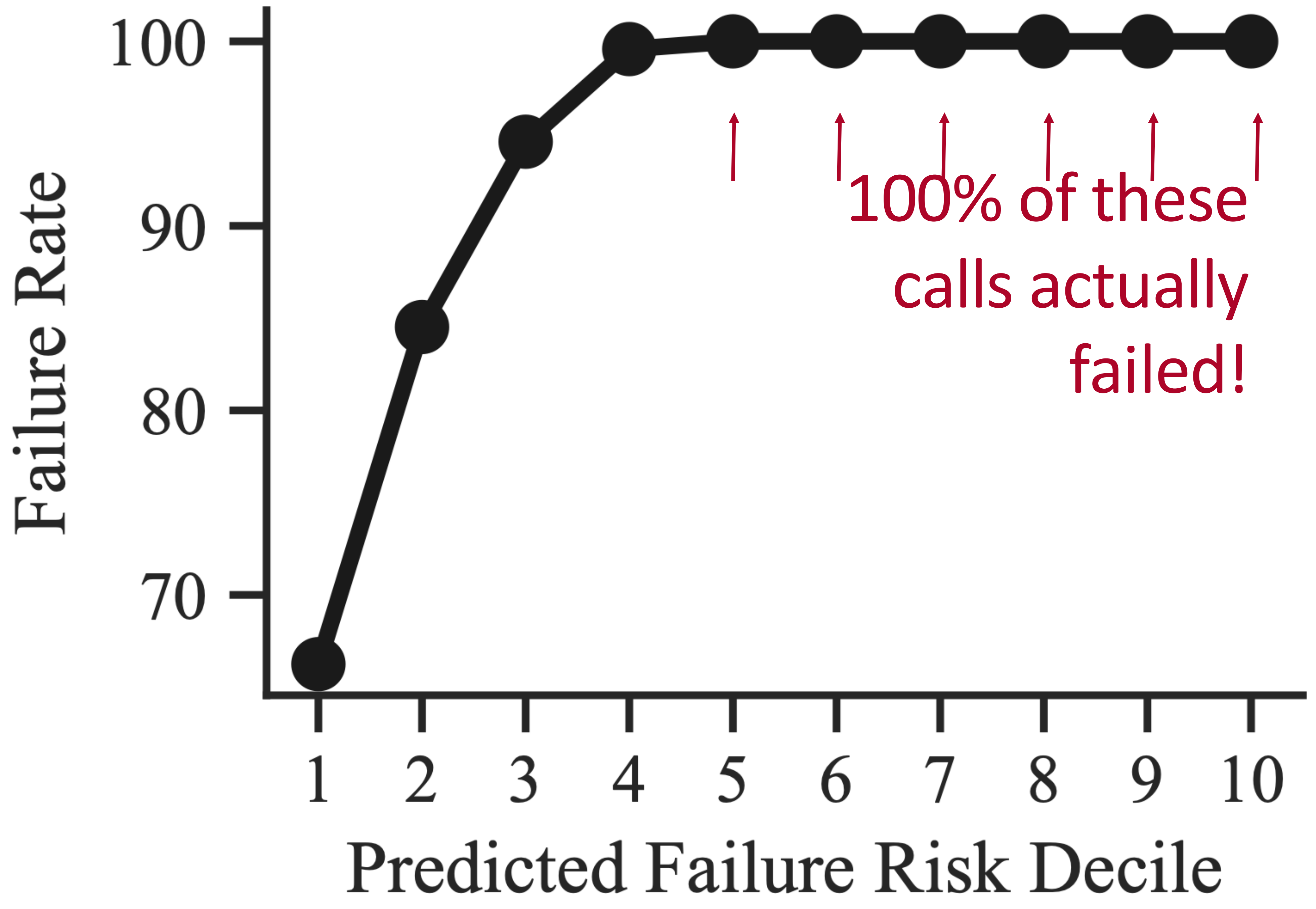
N

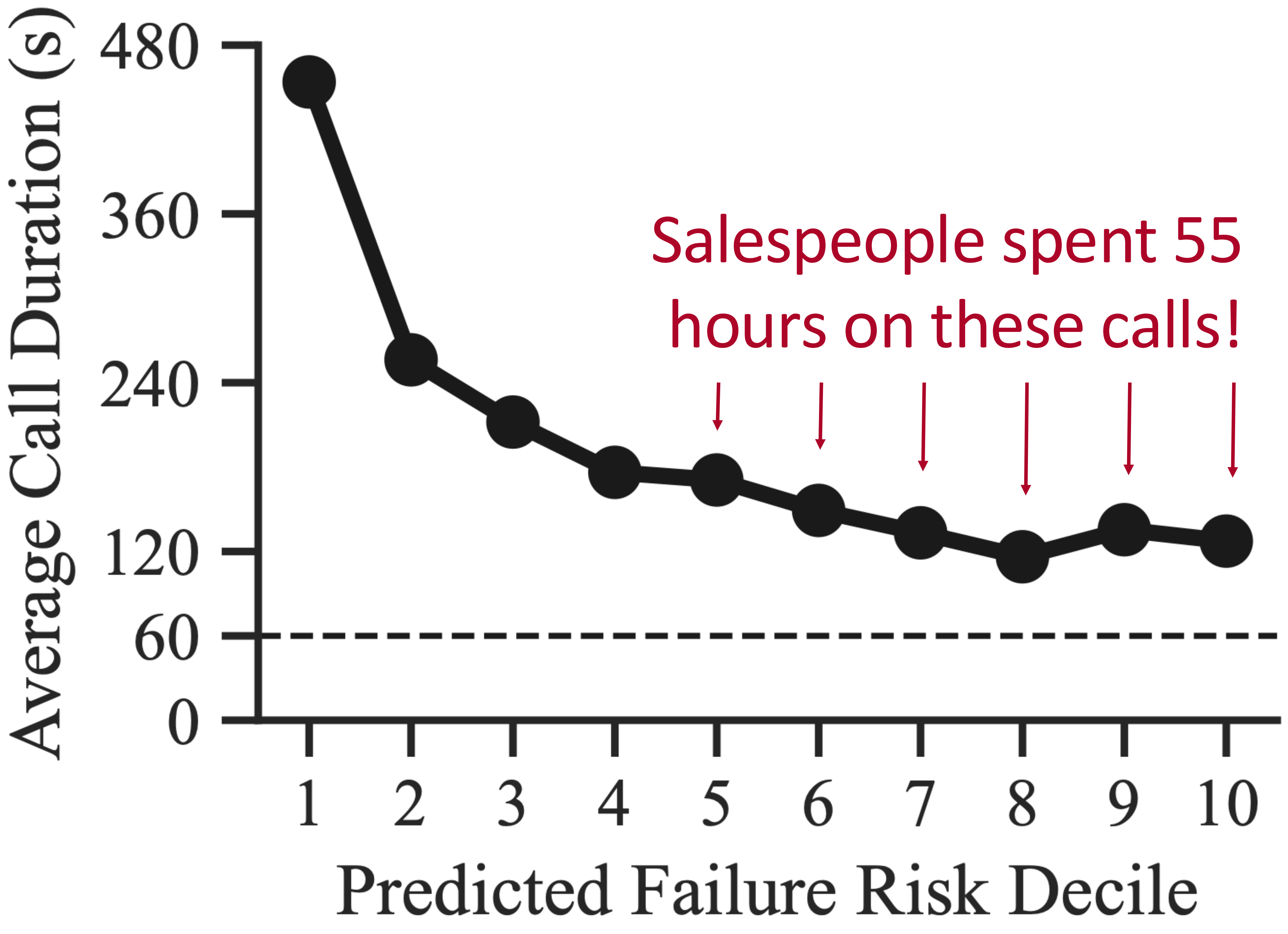
- Create a subset of training calls (50%) and their transcripts (first 60 seconds)
- Wrap transcripts in a prompt
- Set correct response to yes or no based on if the call ended in a sale
- Fine-tune GPT 4.1 to minimize the correct response generation loss

Fine-tuned GPT 4.1 predicts failure risk ($\mathbb{P}[no| \dots]$) 60 seconds into the call

Do calls have early indicators of failure?

Applying the fine-tuned GPT-4.1 predictor to the 2,438 *test* set calls (~20%)





Call failure is predictable early — but when to act on these predictions?

Problem Definition

Defn 1. Policy $\pi_\theta(a_t|s_t)$: Action a_t ∈{quit, wait (do nothing)} given state s_t

State s_t ≡ Transcript text until t

Orador 1 | bueno pues

Orador 1 | imagino que estamos hablando con su senora

Orador 0 | si si si

Orador 0 | pues eso ahorros todos que quieras

Orador 0 | ponlos todos que quieras

$\pi_\theta(a_t|s_t)$ = quit ⟶ Call ends

Problem Definition

Defn 1. Policy $\pi_\theta(a_t|s_t)$: Action a_t ∈{quit, wait (do nothing)} given state s_t

State $s_t \equiv$ Transcript text until t

Orador 1 | bueno pues

Orador 1 | imagino que estamos hablando con su senora

Orador 0 | si si si

Orador 0 | pues eso ahorros todos que quieras

Orador 0 | ponlos todos que quieras

$\pi_\theta(a_t|s_t)$ = wait

State $s_{t+1} \equiv$ Transcript text until t+1

Orador 1 | bueno pues

Orador 1 | imagino que estamos hablando con su senora

Orador 0 | si si si

Orador 0 | pues eso ahorros todos que quieras

Orador 0 | ponlos todos que quieras

Orador 1 | de acuerdo

Orador 1 | pero hay alguna permanencia?

Orador 0 | no no no

Modeling Objective

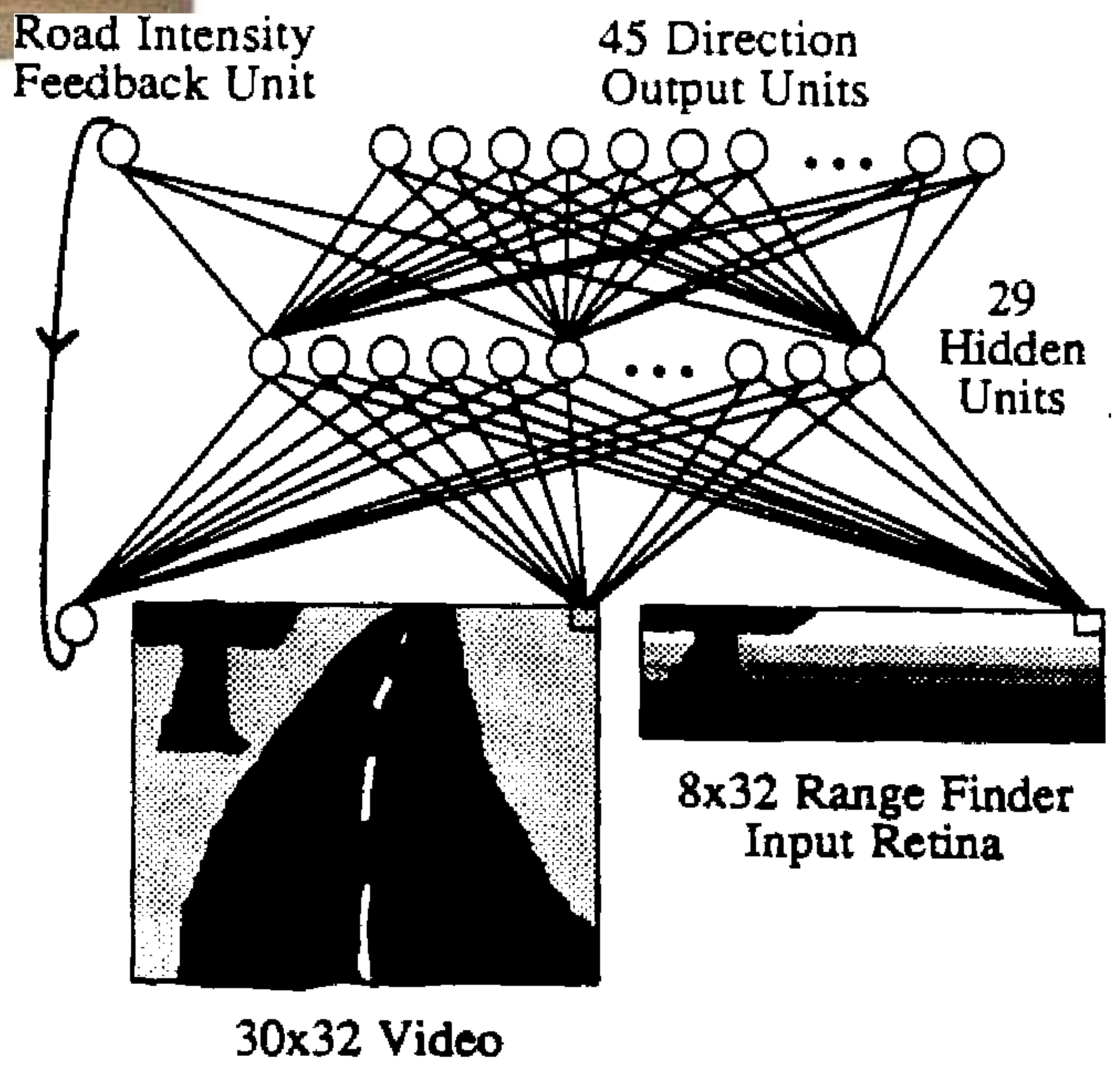
We want to parameterize $\pi_\theta(a_t|s_t)$ with a pretrained generative large language model

- We want a sequential decision policy that fully leverages the language understanding of large, pretrained LLMs.
- BUT, standard approaches to learning LLM policies — via deep reinforcement learning — suffer from instability, hyperparameter sensitivity, and lack of scalability

Our solution: We show that estimating the stopping policy can be reformulated as imitation learning — stable & scalable

Learning via Imitation

ALVINN
(Pomerlau, 1988)



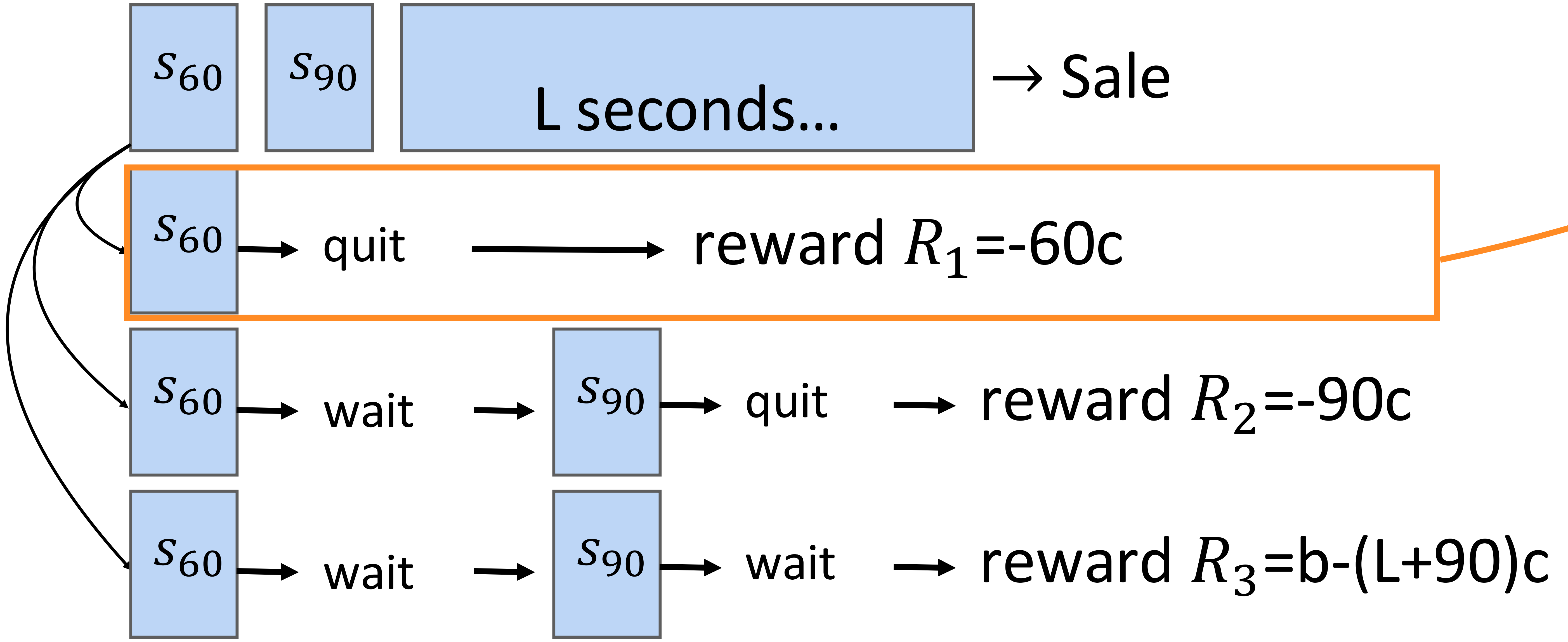
Key disadvantage:

Need “expert” data and a good imitator

Key Insight: Inferring an Expert Policy

Example: Let T=2, waiting cost be c per second & benefit be b per sale

Calculate the cumulative reward for every quitting time τ



Say $R_1 > max(R_2, R_3)$

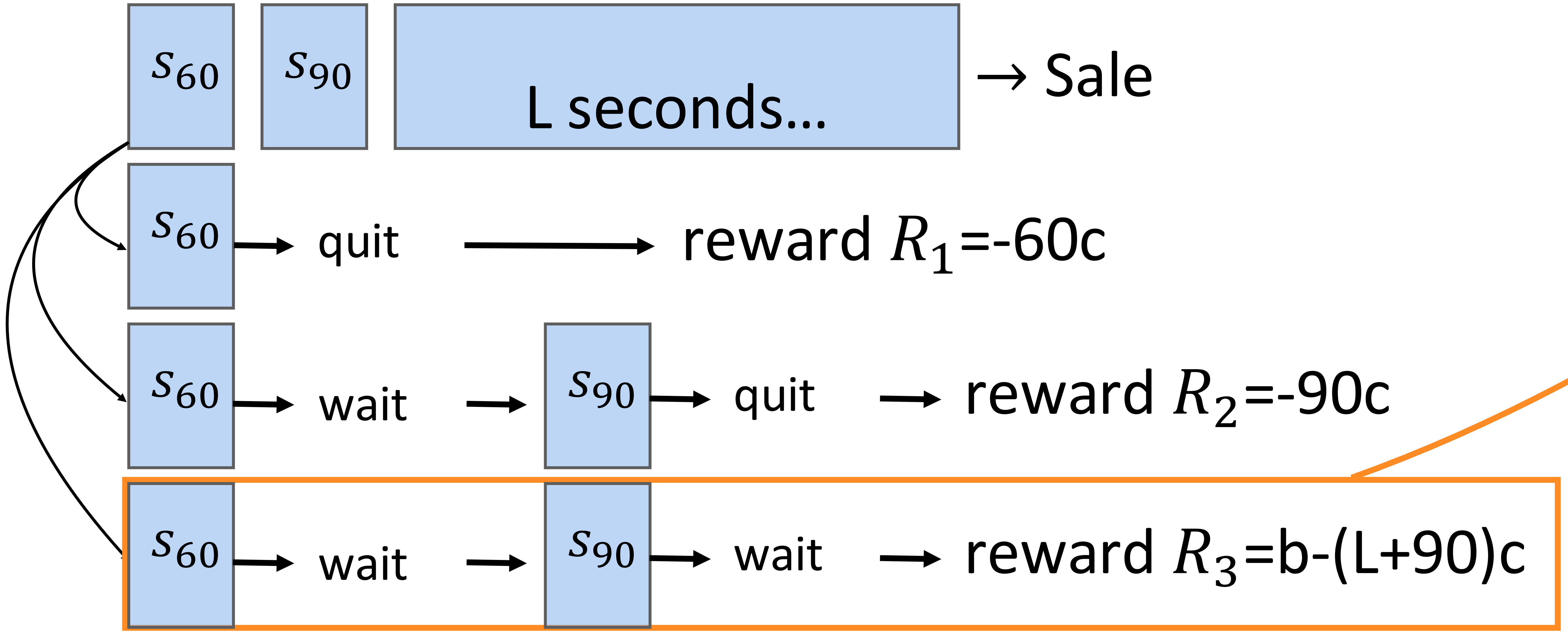
State	Optimal Action
s_{60}	quit

Record the optimal state-action trajectory (that maximizes R_τ)

Key Insight: Inferring an Expert Policy

Example: Let T=2, waiting cost be c per second & benefit be b per sale

Calculate the cumulative reward for every quitting time τ

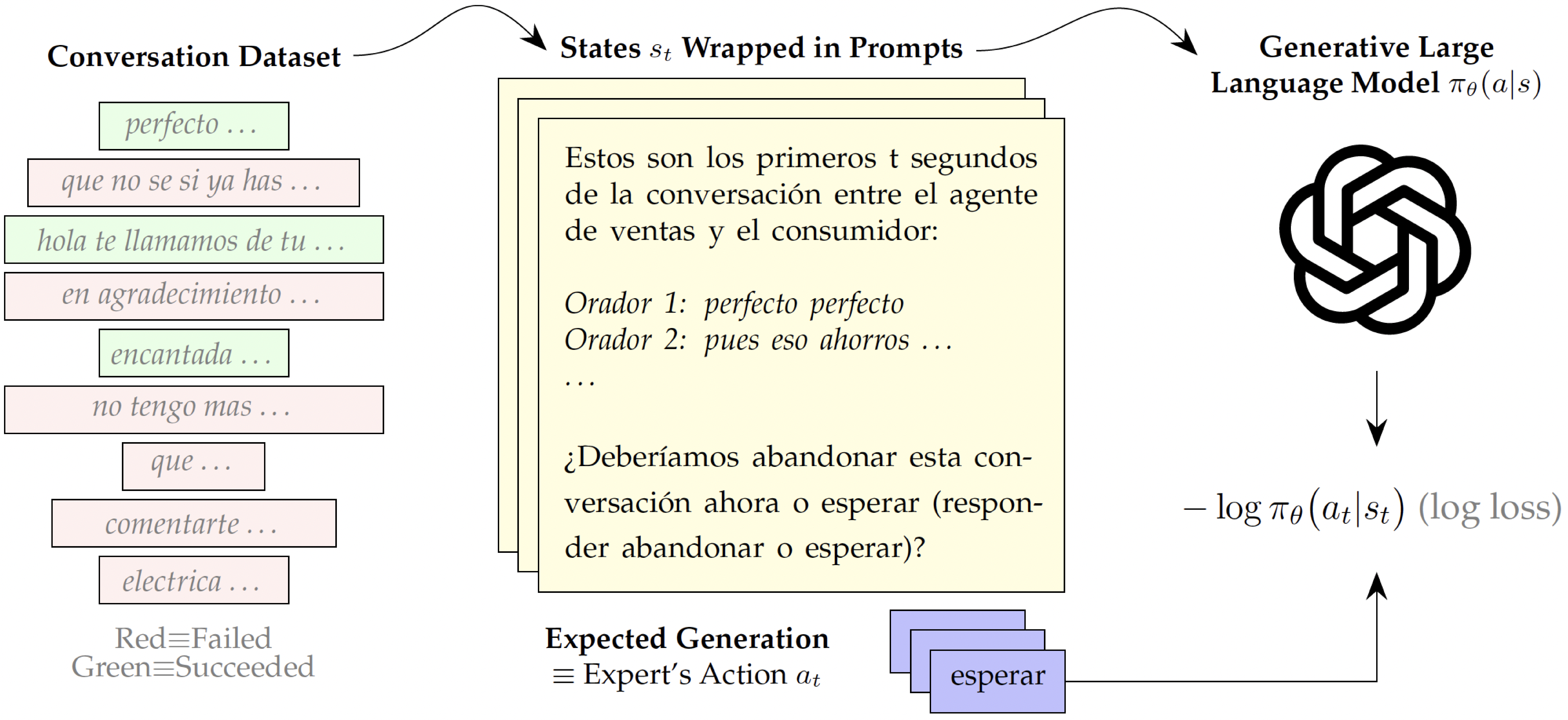


Say $R_3 > max(R_1, R_2)$

State	Optimal Action
s_{60}	wait
s_{90}	wait

Record the optimal state-action trajectory (that maximizes R_τ)

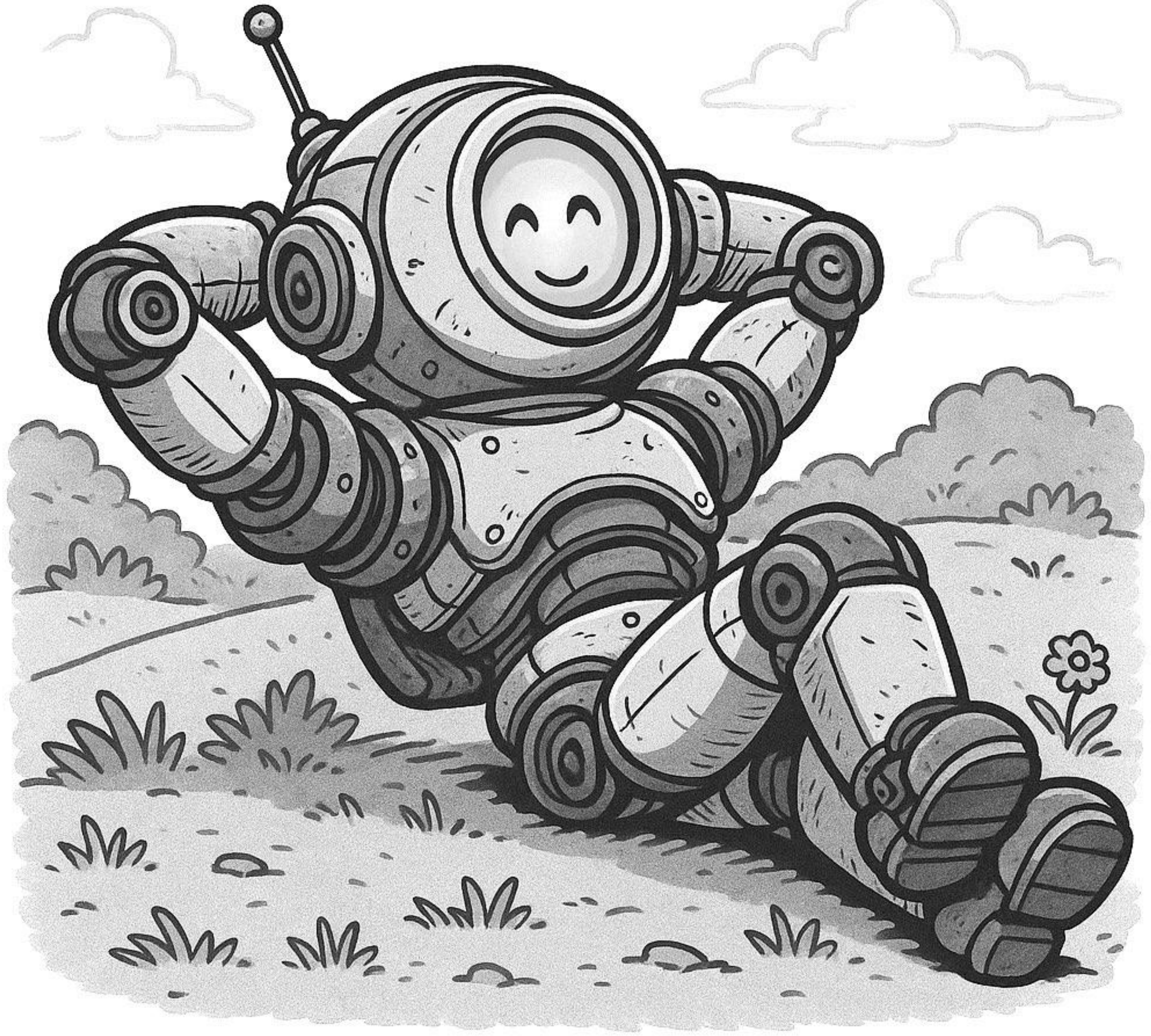
In a Nutshell...



So what have we gained?

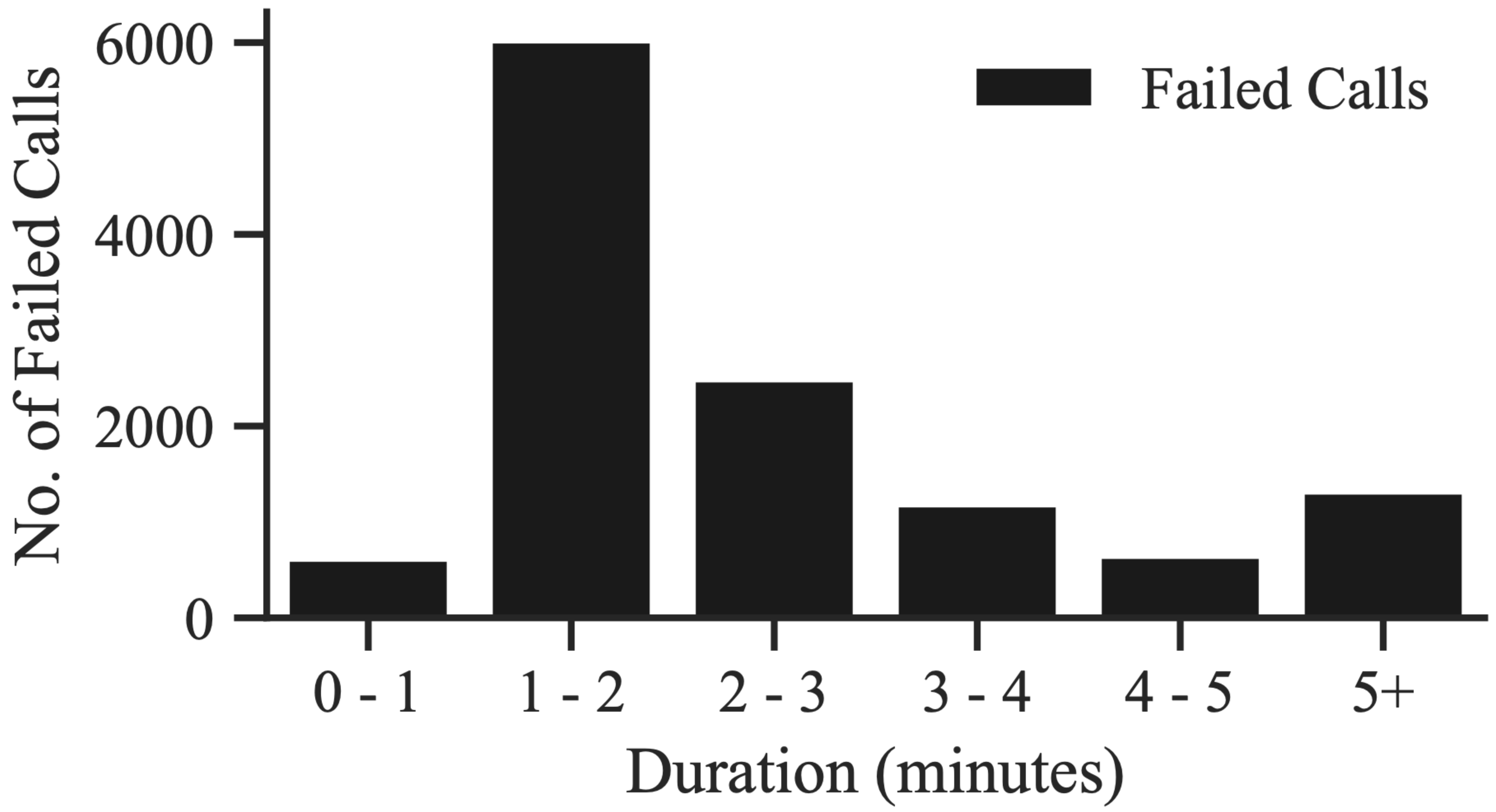
A sequential decision policy over textual states that:

- Is practical & scalable to estimate: Stable training, robust to hyperparameters, inherits scalability and tooling of language model fine-tuning
- Flexibly accommodates various LLMs: Can be used with proprietary language models locked behind restrictive APIs, and open-source ones
- Scales with technology and data: Easily adaptable to future vision, audio, and multimodal models when such data becomes available



Empirical Application

- 11,627 outbound cross-selling calls (first contact) by 79 salespeople at a large European telecom firm over 30 days
- 5.5% success rate, average call is 195 seconds long, failed call is 169 seconds long, successful call is 630 seconds long
- Each salesperson works 2 shifts, makes 147 calls on average over the month, has a success rate of 6.6%

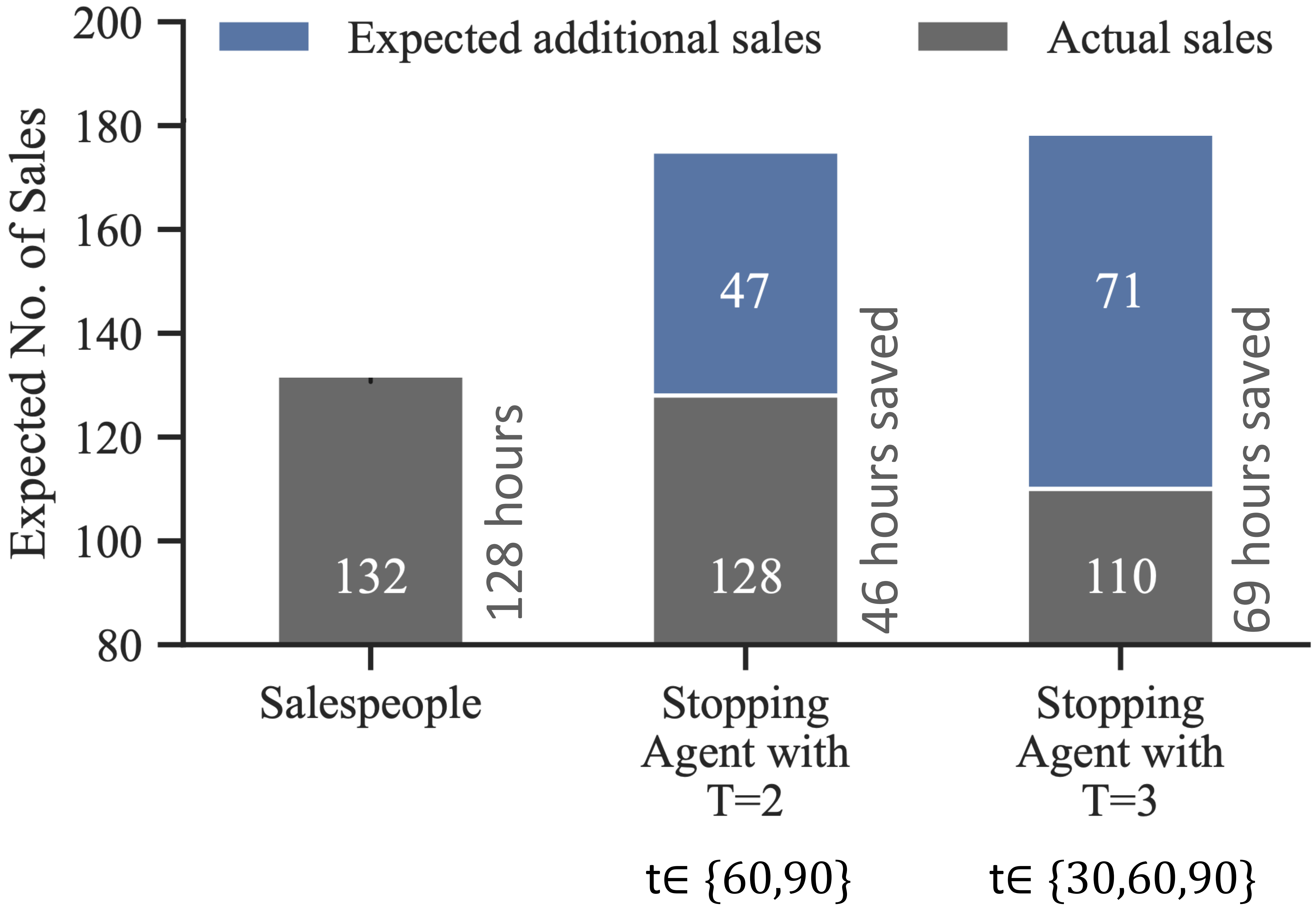


- We hold out 2,438 calls (≈ 20%) as a test set, 3,499 calls for validation (≈ 30%), and train on the remaining (≈ 50%)

Evaluation

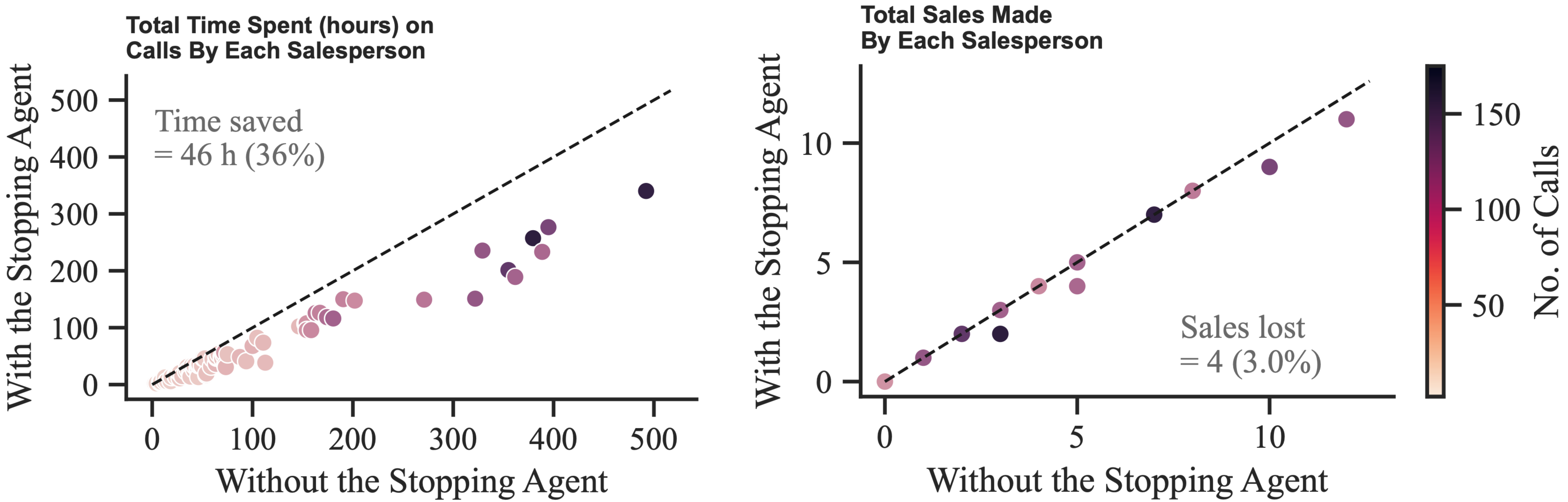
- Two GPT-4.1 stopping agents:
  - T = 2 quits at t∈ {60,90}
  - T = 3 quits at t∈ {30,60,90}
- Our stopping agent only truncates calls ⟹ we can exactly calculate time saved Δt, and the sales lost
- We translate the time saved Δt into expected additional sales:

$$\frac{\Delta t}{Avg.\ Call\ Duration} \times Success\ Rate$$



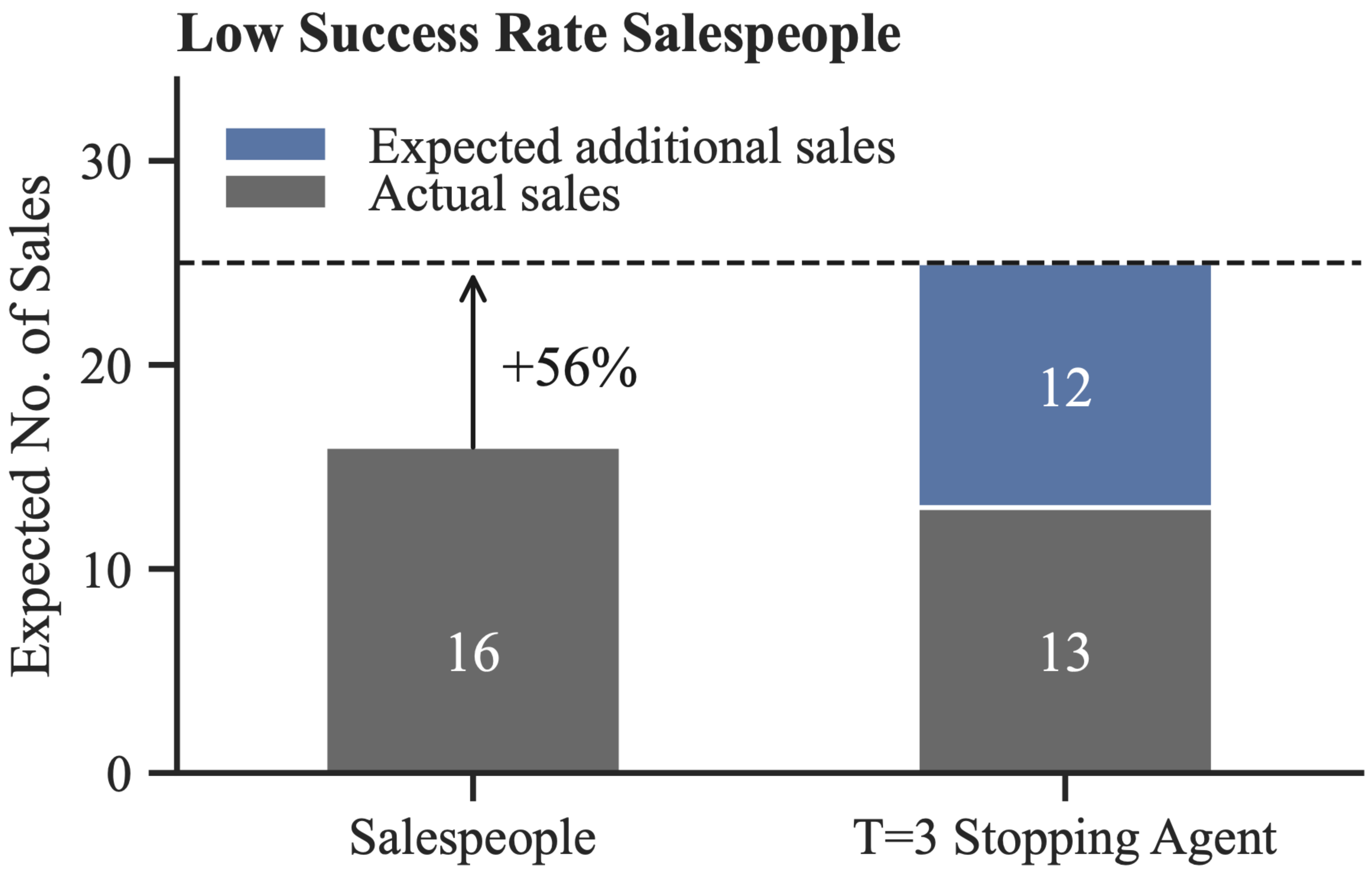
Gains per Salesperson for T=2

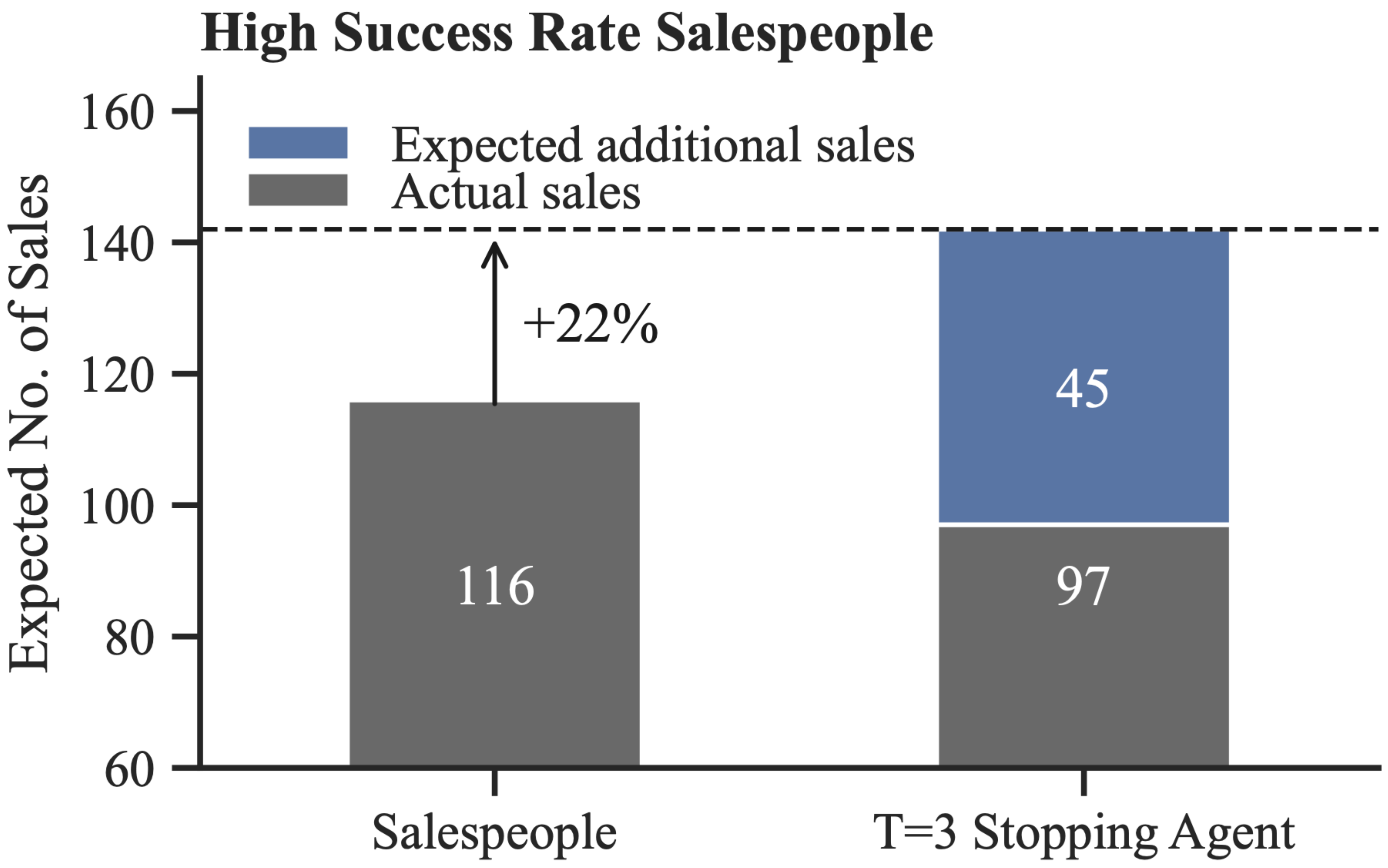
T=2 decision opportunities: at t=60 and at t=90



Heterogeneity with Salesperson Effectiveness

We observe sales gains in different salespeople subgroups without retraining

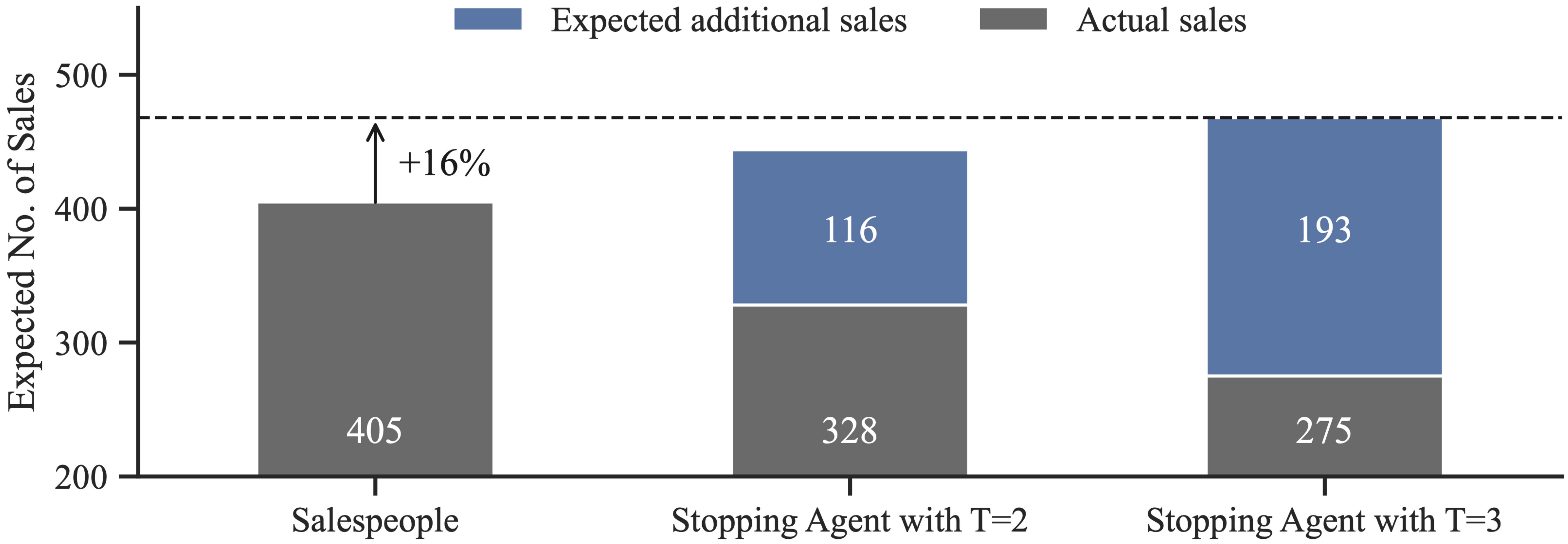




Note: Expected sales calculated using subgroup-specific average durations and success rates

Effectiveness in *other* campaigns

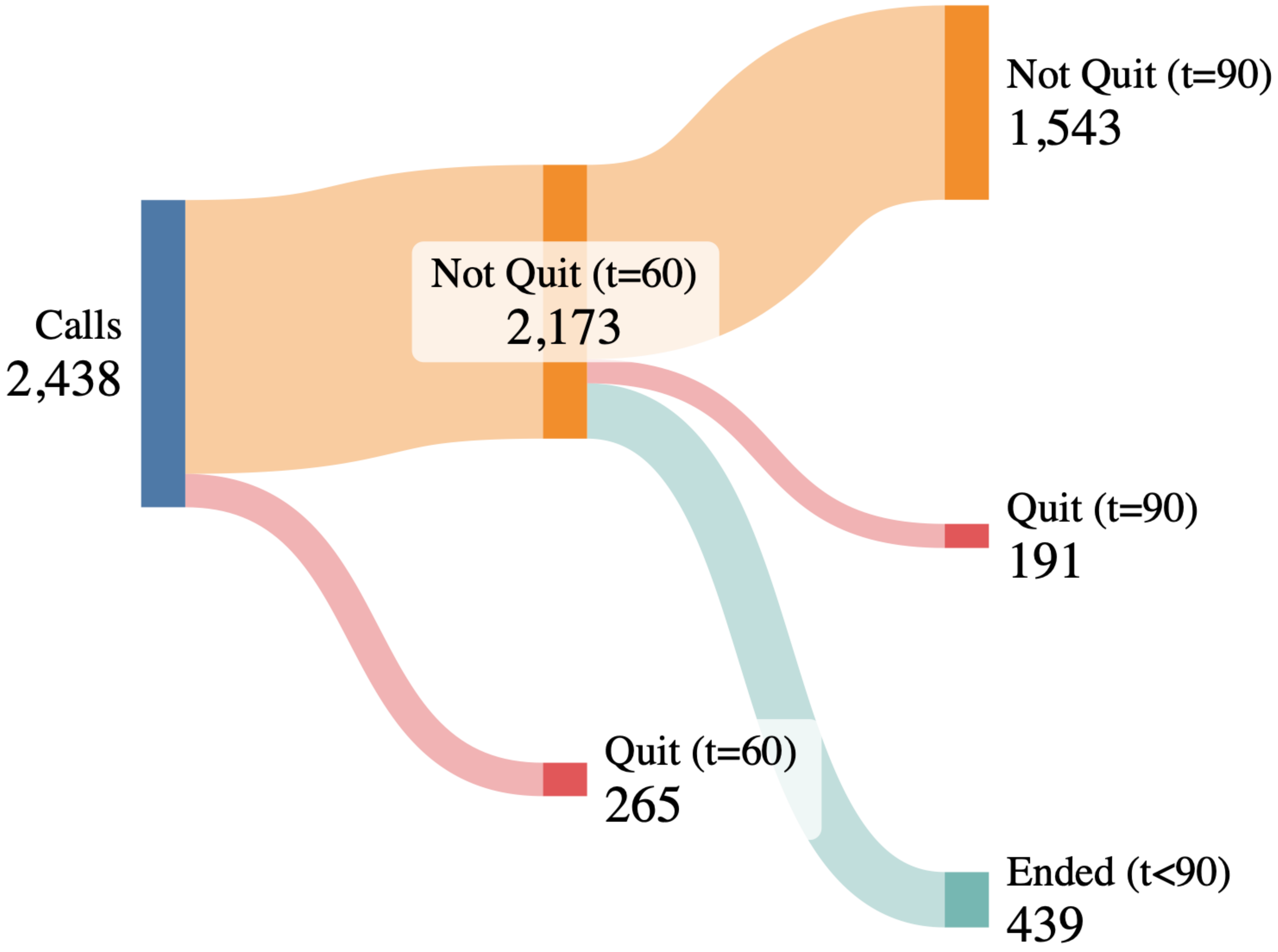
New campaign that targets consumers of a different sub-brand. Only 31 salespeople overlap with the original campaign. Without retraining

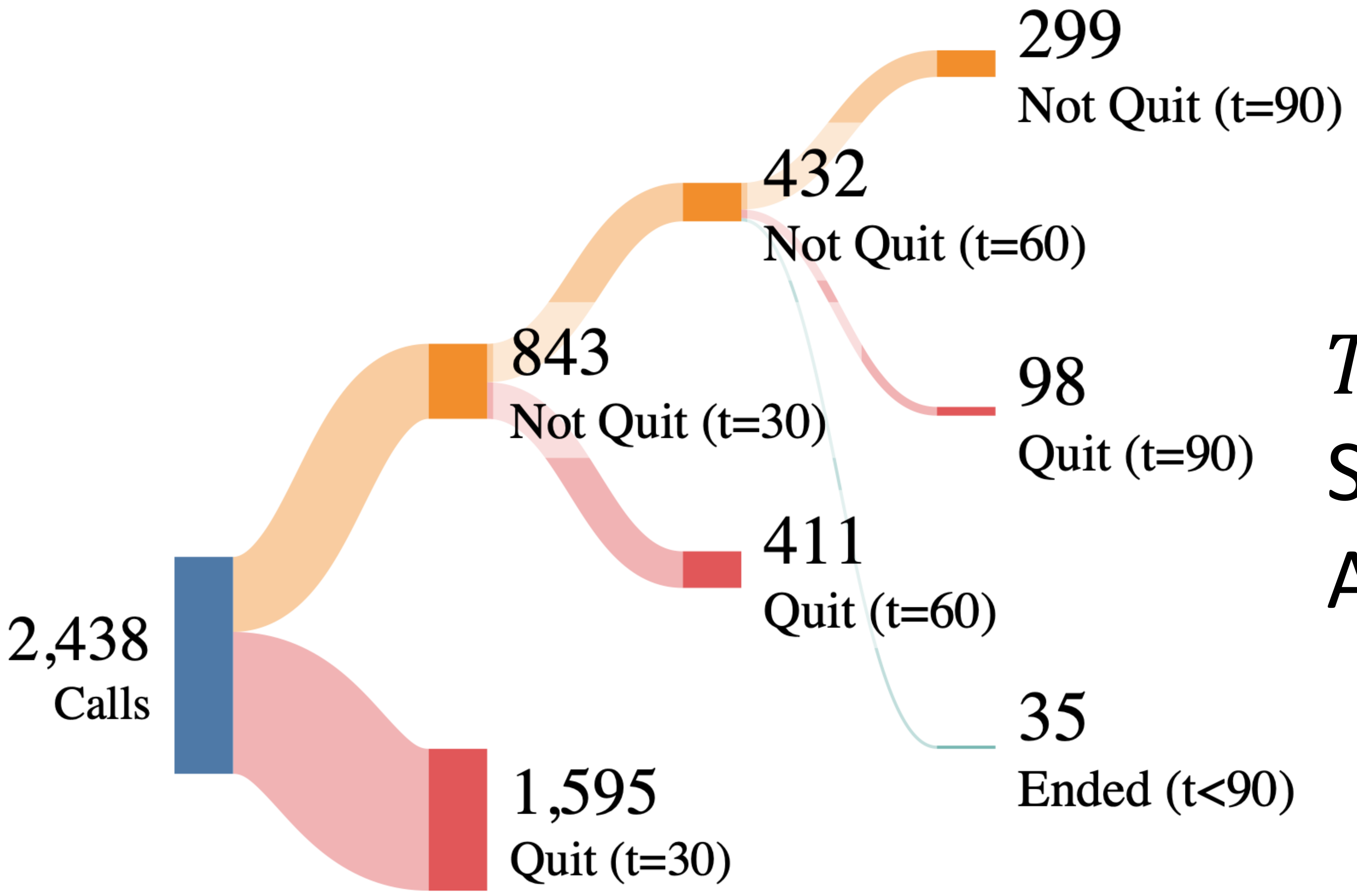


The Stopping Agent is Robust to Distribution Shift

When do salespeople quit?

Salespeopl
e





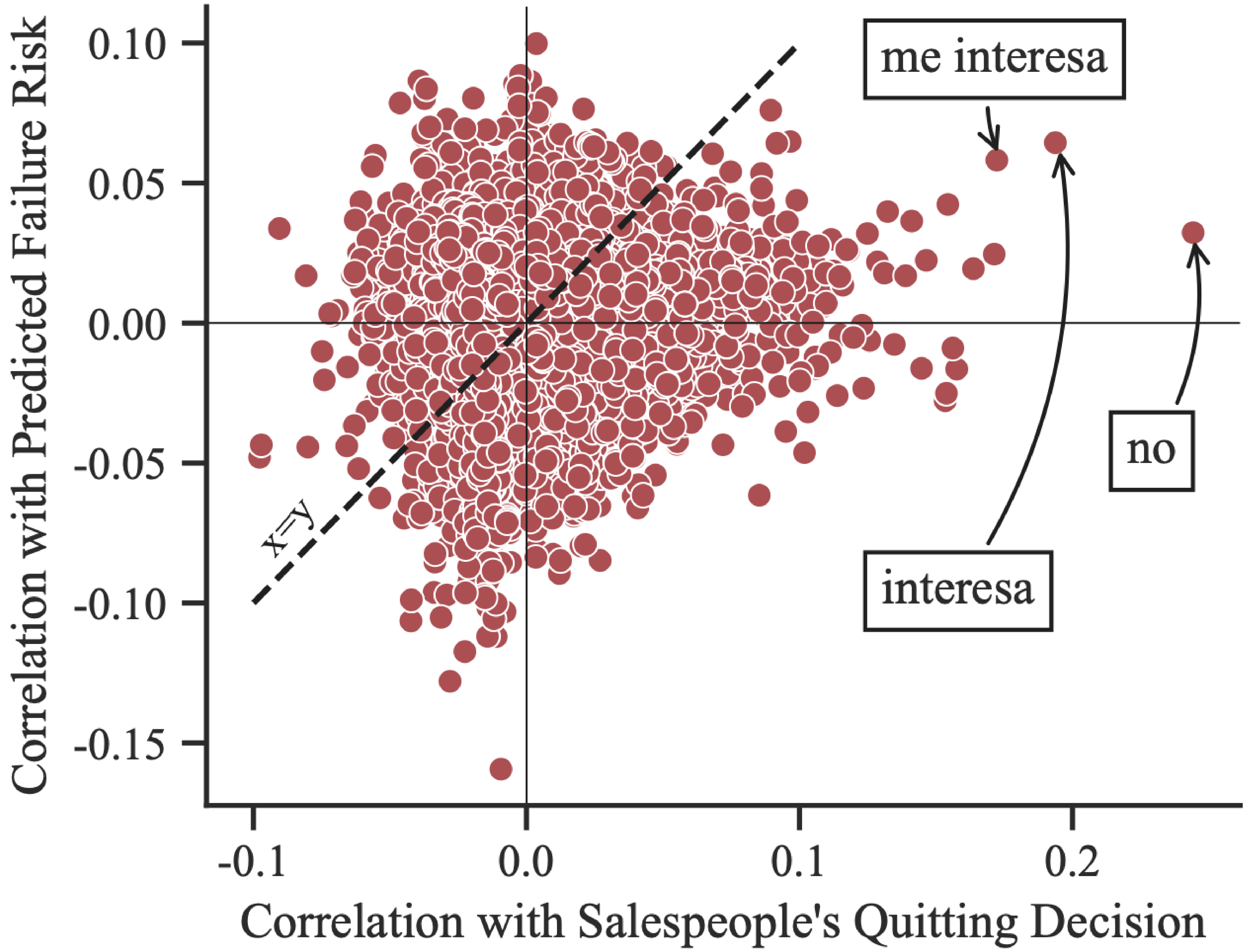
$T = 3$
Stopping
Agent

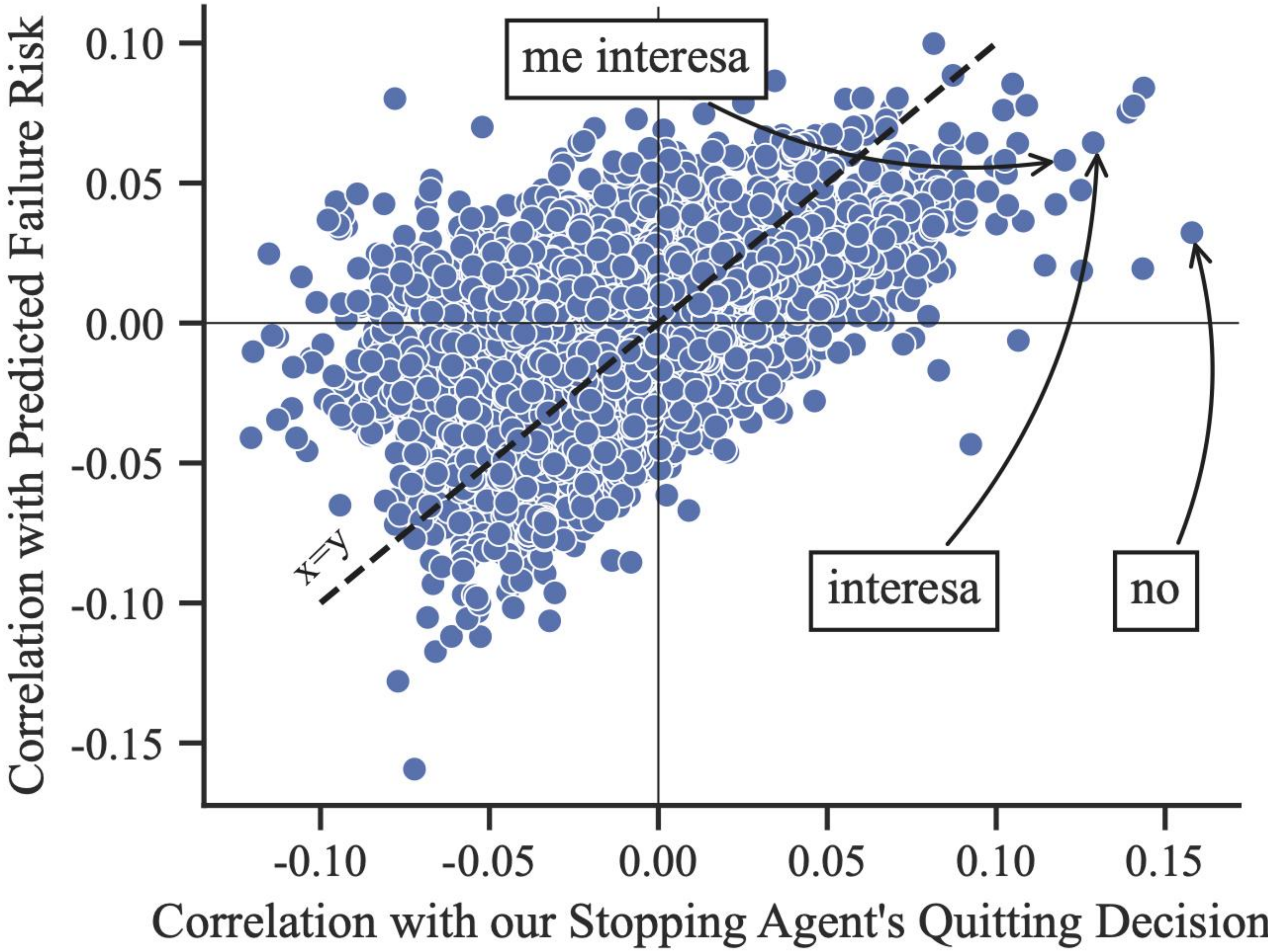
Now we observe humans and AI agents, so we explore which words predict…

1. Quitting decisions
2. Call outcomes

What predicts *quit* vs. *call failure*

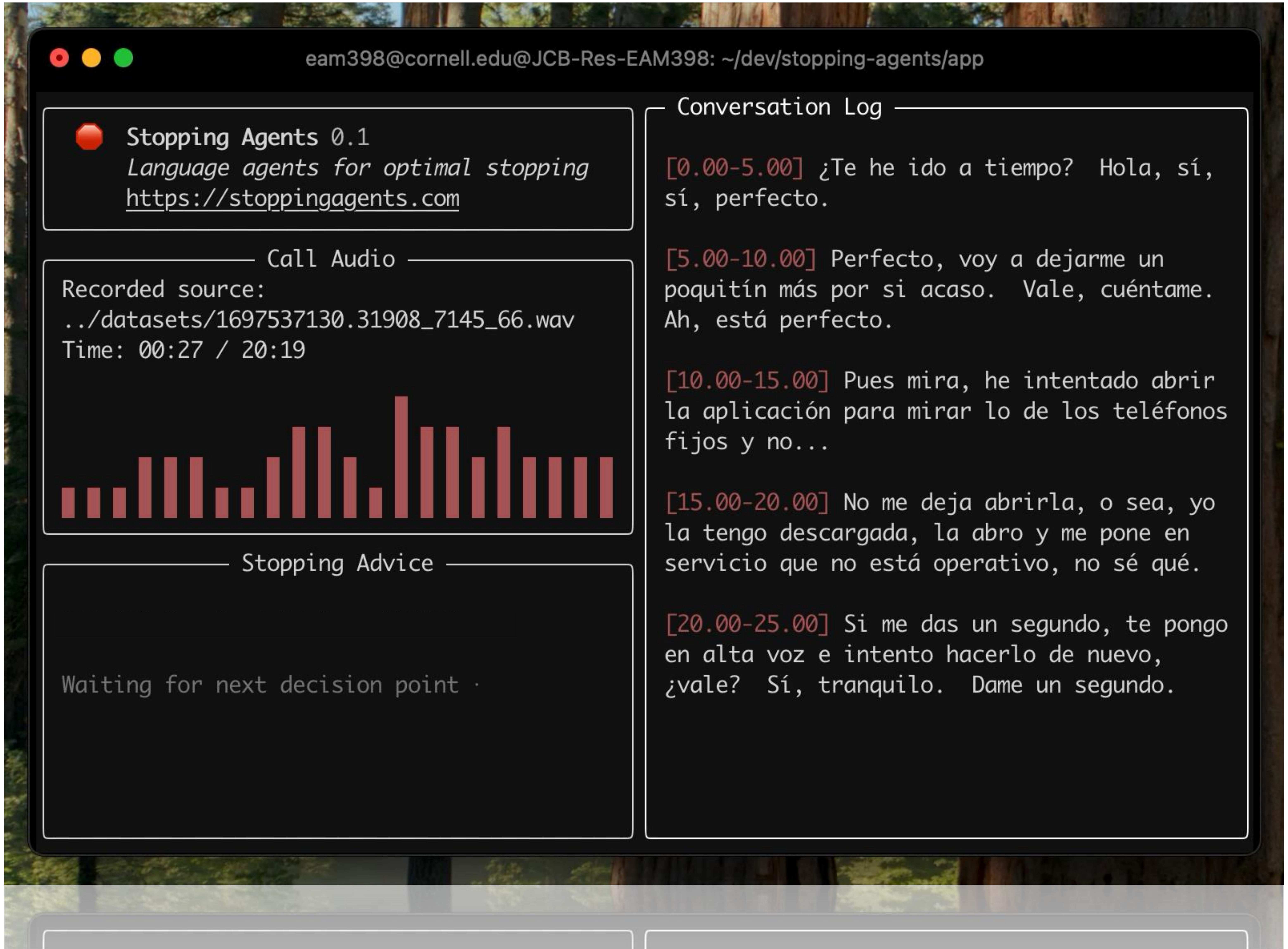
Correlation of unigrams and bigrams with predicted failure risk at t=60





Outsized association of “no me interesa” with quitting relative to predicted failure risk

Demo



- Stopping agent with decision points at t=15, 30, 45 seconds
- Makes real-time decisions given the automatically-transcribed sales call
- Can be embedded in CRMs (e.g., Gong.io)

stopping
agents.co
m

Conclusion

- We propose stopping agents: Language agents for optimal conversation stopping
  - A general method for more efficient human-human communication
- We show 54% time savings $\Longrightarrow$ expected sales gains of 37% on sales calls in the field
- Scalable, cost-effective, usable with both proprietary and open-source language models
- Suggestive evidence of cognitively-bounded salespeople — incentive alignment and contracts alone may not be sufficient to address inefficient disqualification

stoppingagents.com

Thank you!