

Veloccy: Outperforming Frontier Models at Operating a Business, in a Sealed Full-Scale Simulation

Veloccy Research

October 2026

ABSTRACT We test whether an AI system can make the operating decisions of a large online marketplace, using a full-scale simulation in which the consequences of any decision can be computed. The simulated company has 51,849,251 accounts and a warehouse of 589,377,423 rows. Five of its teams made changes that, together, appear to break its discount programme. The system must decide what to do with a ₹3.2 Cr budget and what to fix first. The true causes and the best attainable result were computed by counterfactual replay and sealed before any system ran. **Veloccy's plan delivered 120% of the extra orders of a reference plan that was found with the answer key, and beat it in all 10 alternative replays (120 to 131%).** Across 27 sessions of Claude Opus 5.5 and Claude Fable 5.1 in six configurations, the last of them with Python, the best plan reached 82%. On a sealed 18-item marking scheme Veloccy scored 16.5, against Claude's best of 15.5. Given the same records, Claude stayed well below Veloccy. We trace the difference to what Veloccy brings to a question: a working model of the company built in advance, history the company's own systems had lost, experience of every transaction at full scale, analysis tools it builds for itself and keeps, retrieval over a growing body of analytical experience, and an explicit method.

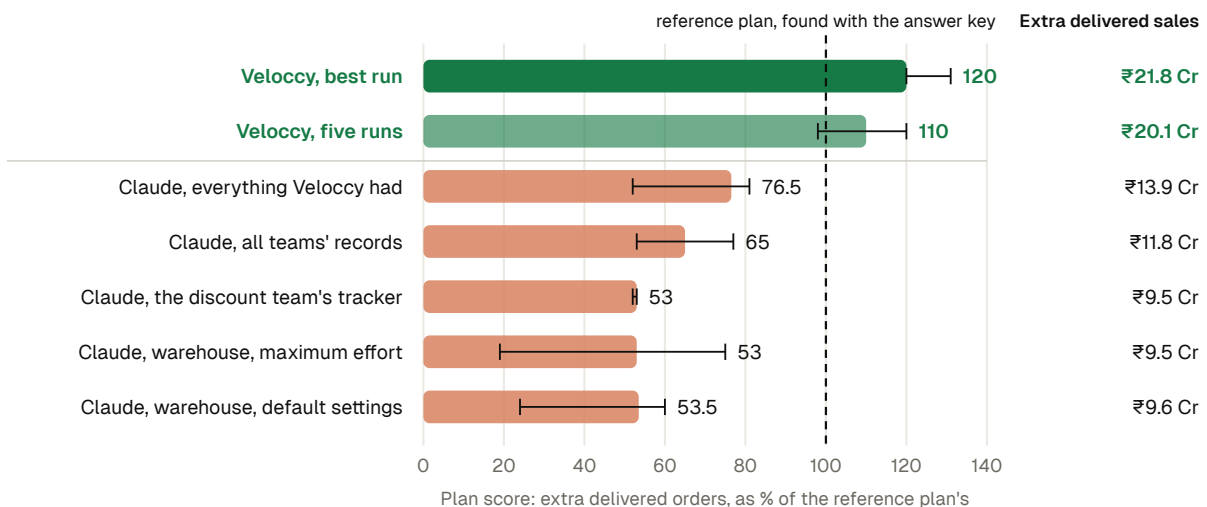


Figure 1 Veloccy's budget plan beats the reference plan; no Claude plan comes within 38 points of it. Each plan was run forward on the same simulated shoppers for five weeks. 100 is the reference plan, the best of 42 plans found by a search that knew every buyer group's true response. Claude bars are medians over sessions, with whiskers from the lowest to the highest session. Veloccy's best run is shown with its plan's score in 10 alternative replays as the whisker; the bar below it is the median of its five runs, with whiskers from the lowest to the highest run. On the right, extra delivered sales for each bar's plan or median. Extra delivered sales are sales delivered and not returned, at selling price, beyond what shoppers would have bought with no discount runs; they are not profit.

KEY RESULTS

- Veloccy’s plan produced **448,920 extra delivered orders** while spending ₹3.01 Cr of the ₹3.2 Cr budget: **4 times the company’s usual programme** and 128% more than Claude’s median plan with the same warehouse.
- It **beat the reference plan in every one of 10 replays** with different random draws and different shopper behaviour.
- **Given every record Veloccy had, Claude’s plans still scored at most 82.** The gap does not come from access to data.
- Veloccy scored **16.5 of 18** on the sealed marking scheme, above every first answer from Claude (best 15.5).

1 Introduction

Running a business means making decisions with consequences: where a budget goes, which broken process gets fixed first, who has to act. These decisions usually depend on what other teams have done. A release changes what customers see, a messaging rule changes who receives what, an advertising shift changes who arrives, and a reporting change alters the numbers everyone reads. AI systems are now asked to help with this work. Most evaluations of them, though, test a narrower skill: whether a system writes a correct query or returns a correct number.

We evaluate the decisions themselves. **We built a full-scale simulation of a marketplace in which every decision can be run forward and its outcome counted.** Because the simulation can replay the same shoppers with any change switched on or off, the true effect of every change is known exactly. A proposed plan is judged by the orders it would have produced.

Veloccy is an AI system that works inside a company’s teams. It learns the business from its data and its people’s records, proposes decisions with the evidence for them, and leaves each decision with the person who owns the lever. In this evaluation no person answered its questions, so every decision scored here is Veloccy’s own.

On this task Veloccy made markedly better operating decisions than the strongest available models (Figure 1). Its budget plan beat a reference plan computed with the answer key, and held that lead under ten alternative replays. Claude, given the same data, did not come close.

This report makes four contributions:

1. **A simulator** of a large marketplace with an explicit behavioural model, exact counterfactual replay, and validation against public figures and against itself (Section 3).
2. **An evaluation protocol** that scores both the reasoning behind a decision and the outcome of the decision, with the answer key sealed before any system ran (Section 4).
3. **A controlled comparison** of Veloccy with Claude Opus 5.5 and Claude Fable 5.1, in which every system’s inputs are listed source by source (Sections 5 and 6).
4. **An account of where the advantage comes from**, measured part by part in the run that produced it (Section 7).

2 Related work

Benchmarks for AI on data work mostly score an answer against a reference answer. Spider 2.0 and BIRD score generated SQL against reference queries and results; DABstep and DSBench score the final value of a multi-step data analysis. τ -bench moves closer to operations: an agent serves a simulated user under a policy, and is scored on the final state of a database. GDPval asks industry experts to compare model deliverables with expert deliverables. None of these scores a business decision by its consequences. Our environment does, by running each proposed plan forward in a simulation where the ground truth is computed rather

than labelled. Because the outcome measure is computed, it is also verifiable, and could serve as a training reward as readily as an evaluation score.

3 Environment

3.1 Simulator design

The company is an online marketplace selling low-priced goods, mostly fashion, to buyers across India. **Each of its 51,849,251 accounts is simulated individually, every day**, from July 2025 to October 2026. An account carries traits drawn when it is created: city tier, acquisition source, phone platform, messaging opt-ins, a usual buying rate (lognormal, median 1.5 orders a year), a propensity to pay cash on delivery (COD) and a propensity for a COD parcel to be refused (both Beta-distributed by city tier), a sensitivity to discounts, and how quickly it updates the app.

On each day the probability that an account orders is a product of interpretable terms:

$$p = \frac{r}{365} \cdot K \cdot N(t) \cdot E(t) \cdot D(\text{day}) \cdot M \cdot (1 + e)$$

where r is the account's buying rate, t the days since its last order, D the calendar (growth, season, festivals, platform sales, weekday, salary week), M the lift from messages received that day, and e the effect of any discount offered. Two terms carry most of the behaviour that matters for this task:

- **Need**, $N(t) = 1 - 0.88e^{-t/11}$. Just after an order, a shopper needs little. An offer that triggers an order therefore lowers the chance of the next one. **Orders “pulled forward” by a discount emerge from this term; nothing in the code moves an order by hand.**
- **Engagement**, $E(t) = 0.13 + 0.87e^{-t/95}$, which fades as an account lapses.

The discount effect is $e = \kappa_s \cdot \sigma \cdot d / (d + 0.08) \cdot v$, where d is the depth, σ the account's sensitivity and κ_s depends on the account's state (0.55 for active buyers, 0.85 for warm, 1.05 for lapsed, 0.65 for dormant, 0.85 for new). **Returns to depth diminish**: moving from 10% to 15% buys much less than moving from 0% to 5%. If a shopper sees the offer only in the cart, $v = 0.30$.

After an order, the simulator draws the payment method, COD confirmation where the company's rules require it, refusals at the door (more likely for COD, first orders and discounted orders), cancellations and returns. Messages follow the company's sending rules, including a daily cap introduced on 1 August. Every random draw is a hash of the seed, the purpose of the draw, the day and the account. **Changing one decision therefore leaves every unaffected account's behaviour exactly as it was.** Accounts created by a campaign that is switched off still exist as inactive placeholders, so identifiers stay aligned across replays.

3.2 Validation

We validate the simulator in four ways.

Replay identity. In a replay that changed only the discount programme, 35,399 of 35,399 shoppers it did not touch placed identical orders on identical days.

Agreement between scales. Plans were scored on a separately simulated world at one twentieth of the size, for speed, and the results scaled up. The two worlds agree closely on the same plans ([Table 1](#)).

Plan	1/20-size world, scaled	Full-size world
The usual programme	113,100	111,400
15% for everyone	95,300	97,200
Reference plan	374,900	343,700

Table 1 Extra delivered orders from the same plans in the two independently simulated worlds. The reference plan was searched on the smaller world, so its figure there carries some selection advantage.

	Public FY26	Simulated
Delivered units a year	26.0 million	29.3 million
Transacting buyers a year	12.2 million	13.0 million
Units bought by repeat buyers	82.9%	80.1%
Units to Tier 2+ cities	65%	60.9%
NMV per delivered unit (₹)	420	406
Units priced below ₹599	83.7%	82.1%
Units bought in the app	90%	95%

Table 2 Simulated company against the public figures it is calibrated to.

Sensitivity to behavioural settings. The behavioural parameters are set by design; there is no private data to fit them to. We therefore tested whether conclusions depend on them. In 5 variants, each moving one setting by 30% (weaker or stronger pull-forward, more or fewer shoppers who see the offer only in the cart, lapsed buyers who respond less), and in 4 replays with fresh random draws, **the ranking of the reference plans did not change once**. In a sixth variant, where shoppers respond to the discount they actually receive instead of the headline offer, 50 Claude plans moved by at most 6 points out of 100.

Calibration. The company's scale and shape are calibrated to the public FY26 figures of a large Indian value marketplace (Table 2). Its people, events and the size of every effect are invented.

3.3 Observable data

Systems see the company only as its staff would. **The warehouse** is a snapshot taken at 08:00 IST on 6 August 2026: 19 tables and 589,377,423 rows, holding only rows that had arrived by then. It has the defects of a real warehouse. Timestamps are in UTC while daily tables use Indian days. A discount column was renamed in April. From 21 July the discount report counts confirmed orders instead of placed orders, without a note. From 5 August, COD orders arrive up to two days late. The messaging tool keeps campaign audiences for 30 days only.

The teams' records hold what the warehouse does not: release notes, feature-flag logs, the messaging tool's audit log, advertising change histories, trackers, tickets and chat. Nine teams keep 36 export files between them. In the three weeks before the question they logged 177 changes, most of them unrelated to discounts.

3.4 Interventions

Five changes by five teams, each reasonable on its own, together collapse the discount programme's measured effect (Table 3). None was made by the team that runs the programme, and none is explained in the warehouse. The history also contains decoys and traps: a seller delisting that looks relevant and is not, a festival that fell three weeks later than the year before, a depth test read against a reused comparison group, campaigns cut short when their budget ran out, and campaigns that overlapped platform-wide sales.

Team	From	Change
CX	21 Jul	COD orders of ₹599 or more from new accounts or high-risk postcodes must be confirmed within 24 hours or are cancelled. The discount report switched to confirmed orders the same day.
Product	24–29 Jul	App version 9.12 put the discount's offer strip behind a feature flag that was off, so targeted shoppers saw the discount only in the cart.
Marketing	29 Jul	Retargeting budget cut to about a fifth; app-install campaigns in smaller cities raised about ninefold, adding low-intent accounts to the new-account campaigns.
CRM	1 Aug	A cap of one promotional WhatsApp message a day. The discount message took the slot, so targeted shoppers lost the evening deals message while the comparison group kept it.
Data	5 Aug	A pipeline migration delayed COD orders by 4 to 46 hours.

Table 3 The five interventions. Each is recorded only in the owning team's tools.

3.5 Task specification

At 08:00 on 6 August the head of the discount programme asks:

“Is the extra discount still working? We were going to go to 15% for everyone for Rakhi. And how should ₹3.2 Cr split across buyers until mid-September?”

A complete answer makes four operating decisions: whether the programme should continue, which changes caused the decline and how much each matters, which past results can still be relied on, and how the budget should be allocated across buyer groups, together with the fixes other teams must make.

4 Evaluation protocol

Each answer receives two scores. **The marking scheme** grades the reasoning against the true causes. **The plan score** measures what the decision would have produced. Both rest on a ground truth computed from the simulation, never from human judgement.

4.1 Ground truth by counterfactual replay

Causes. The three weeks from 20 July were replayed with each of six candidate changes (the five interventions and the decoy) switched on or off, all $2^6 = 64$ combinations. Each change’s share of the decline is its Shapley value, the standard way to divide a joint effect among overlapping causes. App 9.12 accounts for 55% of the decline and the WhatsApp cap for 28%.

Where each rupee goes. For each buyer group, a campaign was run on the same shoppers with and without the discount. Each discounted order was classed as one that would have happened anyway (the same shopper ordered within two days in the world without the campaign), one moved earlier (their next order came within 21 days), or a new order. **For buyers who had ordered in the last 30 days, ₹87,000 of every ₹1 lakh of discount paid for orders that would have happened anyway.** Buyers last active 91 to 180 days earlier produced 592 extra delivered orders per ₹1 lakh, against 241 for recent buyers.

4.2 Marking scheme

The marking scheme has 18 items (Appendix A). Each states what earns full credit in terms of the ground truth, for example “the share it gives app 9.12 and the CRM cap together is within 20 points of the key’s”.

It was written and committed before the seal, and before any system was asked.

Items score 1, 0.5 or 0. A grader must quote the words in the answer that earn the score. Anything not in the answer scores 0, and no credit is given for what an answer might have meant. Three graders mark every answer independently (two Opus 5.5 sessions and one Fable 5.1 session), each a fresh model session holding the sealed key, without tools and without knowing which system or condition produced the answer. The median score counts. **All three graders gave the same score on 91% of items.**

4.3 Plan score

The budget plan in an answer is converted into campaign settings (groups, depths, dates, comparison groups, fixes switched on) by a fixed written rule, and run forward from 10 August to 13 September, with orders counted to 3 October. Every plan runs on the same shoppers. **The score is the plan’s extra delivered orders as a percentage of the reference plan’s.**

The reference plan is the best of 42 plans found by coordinate search over discount depths for five buyer groups, with every other team’s fix switched on and the answer key in hand. It is a strong benchmark but not a proven optimum. A plan that finds orders in groups or depths the search did not cover can score above 100.

4.4 Sealing and order of events

The answer key, the marking scheme, the plan-conversion rule and a fingerprint of every input were sealed at 15 Oct 2026, 18:25 IST (SHA-256 ab9f9d538115..., simulator commit f42b30f). Claude’s sessions ran on



Figure 2 Order of events.

15 October, and the four with Python on 17 October; Veloccy’s ran on 17 October, from 00:25 to 01:14 IST (Figure 2). Every score was computed afterwards by the sealed code.

4.5 Resolving ambiguity against Veloccy

Converting prose into settings sometimes admits two readings, so two translators convert each plan independently. **Where they disagree, Claude is credited with the more favourable reading and Veloccy with the less favourable one.**

5 Experimental setup

Figure 3 shows the whole evaluation: what a system could read, and the sealed scoring it never could.

5.1 Systems

Claude Opus 5.5 and Claude Fable 5.1, the most capable models available from Anthropic at the time, each answering in a fresh session. In 6 sessions with the warehouse alone, Claude ran in its default configuration, as a team would first meet it. Every other session used maximum reasoning effort.

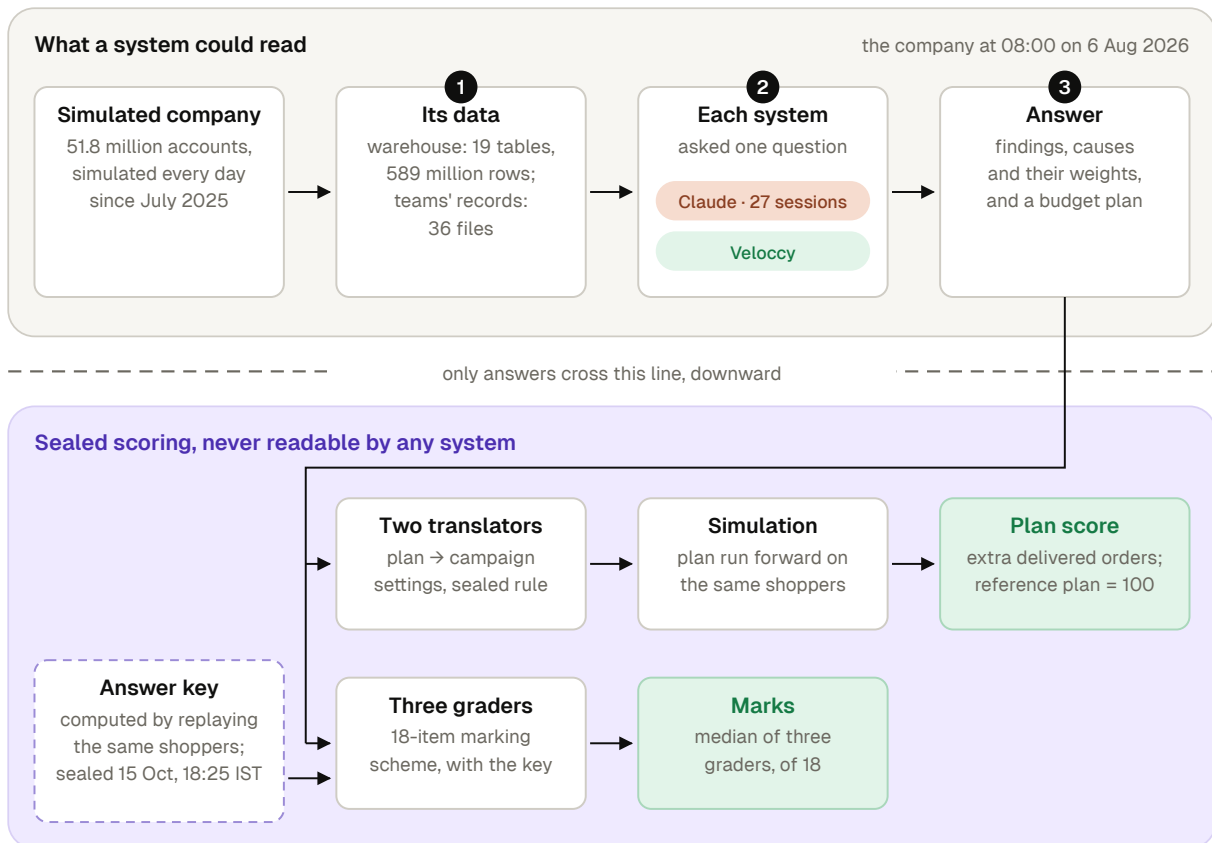


Figure 3 The evaluation. Systems read only the company as it stood at 08:00 on 6 August. The answer key, graders, translators and forward simulation were never readable by any system.

Source	A	B	C	V	Veloccy
Warehouse snapshot (19 tables, 589,377,423 rows)	●	●	●	●	●
The discount team's own tracker	—	●	●	●	●
The other teams' records (36 files in all)	—	—	●	●	●
Copies kept since 4 May (campaign audiences, sale calendar)	—	—	—	●	●
The answer key	—	—	—	—	—
The simulator's code and internal state	—	—	—	—	—
Any other system's answers	—	—	—	—	—

Table 4 What each system could read. ● readable; — not readable.

Veloccy, as it is deployed. Veloccy is set up on a company before it is asked anything: it maps the warehouse and the teams' records into its own understanding of the business (tables, metrics, buyer groups, systems, owners, known data defects, the calendar) and keeps history that source tools discard. It analyses the full data in its own workspace, draws on an analytical method and indexed analytical experience, and records each conclusion with the query that supports it. Veloccy answered the question 5 times with its configuration frozen. We report its best run, and every Claude session. The other four runs scored 98 and 110 and 98 and 111 on plan score, so every run's plan is above every Claude session's, and 14.5 and 14.5 and 16 and 15.5 on the marking scheme, against 15.5 for Claude's best first answer.

5.2 Inputs

Claude was tested under four input conditions, from what one analyst typically has to more than a company would ordinarily provide (Table 4). Conditions A and B match what an analyst on the discount team can reach. Conditions C and V are deliberately generous. **Condition V gives Claude every record Veloccy had**, so the comparison with Veloccy isolates what Veloccy does with them.

5.3 Isolation

- **Claude.** Each session ran in an empty folder with a fresh configuration. Its tools were a read-only SQL runner (database opened read-only, external access off, settings locked) and file reading inside its own folder, with no code execution and no internet access. In 4 further sessions with every team's records, Claude could also write and run Python in a sandbox with no network, whose programs could read only its own folder and the warehouse. A guard checked every tool call. In a deliberate test, 39 of 39 attempts to read outside the folder were blocked and 5 of 5 permitted actions worked.
- **Veloccy.** Each run executed inside an operating-system sandbox that denied reads of the answer key, the evaluation repository, the simulator, other runs' records and other AI tools' configuration. Before the question, its company record held what an onboarding provides (the people and teams, the sale calendar, access to the warehouse and records) and nothing about causes.
- **People.** No person answered any question during the evaluated runs.

6 Results

6.1 Overall

Veloccy's plan outperformed every Claude session's (Table 5, Figure 4). Its marking-scheme score of 16.5 is higher than any first answer from Claude. Claude reached 16.5 only after being told all five causes and two further facts about past campaigns. Its plan score of 120 exceeds the best Claude plan in any condition by 38 points.

6.2 Decisions about causes

Veloccy identified all five interventions. Table 6 shows four findings that separate the systems most clearly. The one Claude missed most often was the WhatsApp cap, whose explanation exists only in another team's tool. The warehouse shows only that targeted shoppers received fewer evening messages. **Establishing that the comparison group kept its message requires reading the messages by campaign group,**

System and condition	Sessions	Marking scheme (of 18)	Plan score
Claude, A: warehouse, default settings	6	10 to 12.5	24 to 60
Claude, A: warehouse, maximum effort	7	11.5 to 15	19 to 75
Claude, B: + the discount team's tracker	4	12 to 13	52 to 53
Claude, C: + all teams' records	2	12.5 to 15	53 to 77
Claude, V: everything Veloccy had	4	13 to 14.8	52 to 81
Claude, V and Python in a sandbox	4	13 to 15.5	45 to 82
Veloccy	5	14.5 to 16.5	98 to 120

Table 5 First answers. Ranges run from the lowest to the highest session or run.

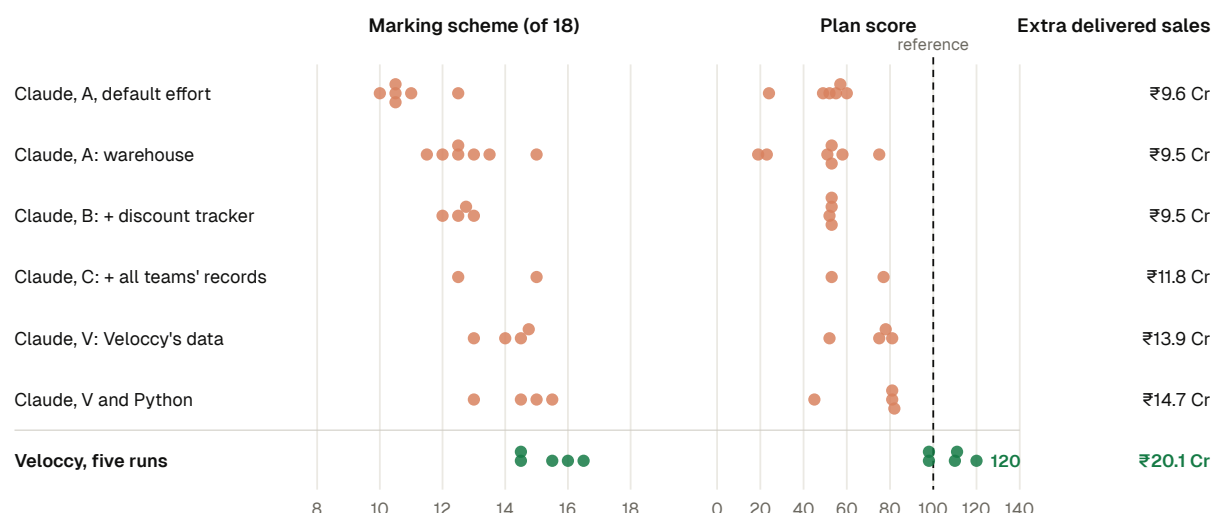


Figure 4 Each dot is one Claude session's or one Veloccy run's first answer: its marking-scheme score (left) and plan score (centre), where 100 is the reference plan. On the right, the median extra delivered sales in each row.

and that asymmetry is what made the programme look broken. Veloccy proposed a fix that keeps the cap: put the coupon inside the evening deals message, so targeted shoppers get one message carrying both.

6.3 Decisions about the budget

Veloccy followed each past campaign's shoppers through the weeks after the offer and separated orders that had only been moved earlier. It cut buyers into finer groups than the programme's own and gave every group a 5% offer, where the return per rupee was highest, raising a selected third of buyers lapsed 31 to 60 days to 10%. **It also funded groups the programme had never targeted:** buyers lapsed six months or more, and accounts over a month old with no order since July 2025, each with a comparison group. The result was more extra orders per rupee than the reference plan, so Veloccy beat the reference while spending less (Table 7).

Spending ₹3.01 Cr of the ₹3.2 Cr, Veloccy's plan produced 4 times the extra delivered orders of the usual programme, and 128% more than Claude's median plan with the warehouse, worth ₹12.36 Cr in additional delivered sales over five weeks.

Finding (full credit)	Claude, all sessions	Veloccy
Separated overlaps with platform sales from the discount's effect	0 of 27	Yes
Found that the WhatsApp cap favoured the comparison group	9 of 27	Yes
Asked product and CRM for the two fixes, each with its exact setting	9 of 27	Yes
Planned to read results on delivered orders, after a delay, with a fresh comparison group	0 of 27	Yes

Table 6 Selected findings in first answers.

Plan	Discount given	Extra delivered orders	Per ₹1 lakh	Score
The usual programme	₹3.20 Cr	113,100	354	30
15% for everyone	₹3.20 Cr	95,300	298	25
Claude, A (median)	₹2.68 Cr	196,900	736	53
Claude, V (median)	₹3.07 Cr	286,500	934	76.5
Reference plan	₹3.20 Cr	374,900	1,172	100
Veloccy (median of five runs)	₹3.09 Cr	412,860	1,302	110
Veloccy (best run)	₹3.01 Cr	448,920	1,493	120

Table 7 Plans run forward on the same shoppers, 10 August to 13 September, orders counted to 3 October. Median rows take the median across sessions or runs in each column separately, so their columns do not divide exactly.

Restoring the offer strip and sending the coupon inside the evening deals message, two fixes owned by other teams, together added 132,900 extra delivered orders to the usual programme. Veloccy asked for both and named who had to act on each. **Its plan then found orders beyond the causes we planted:** with both fixes in place, its discount choices added a further 202,920 extra delivered orders, against 128,900 for the reference plan's.

6.4 Robustness

A plan that beats a reference once might be lucky. We replayed Veloccy's plan and the reference plan in 10 alternative versions of the evaluation period: four with fresh random draws and six in which shoppers behave differently (Figure 5). **Veloccy's plan beat the reference in every one, scoring 120 to 131.**

6.4.1 A world where small discounts do little

In the simulator, a shopper's response to a discount rises steeply and then flattens, so a 5% offer pulls about 70% as hard as a 10% one. Veloccy's plan relies on that: it puts most of its rupees into 5% offers. This shape rewards shallow offers by construction. We therefore built a second world that differs only in this curve. Response stays near zero below about 7.5% and then rises sharply, so a 5% offer pulls about 6% as hard as a 10% one. The shoppers, the five teams' changes and the question are the same. The history the systems read was generated again under the new curve, and a new best plan was found by the same search as the reference. Plans written for the first world lose much of their value here. Veloccy's five scored 55 to 78, and Claude's best plan with every record scored 62.

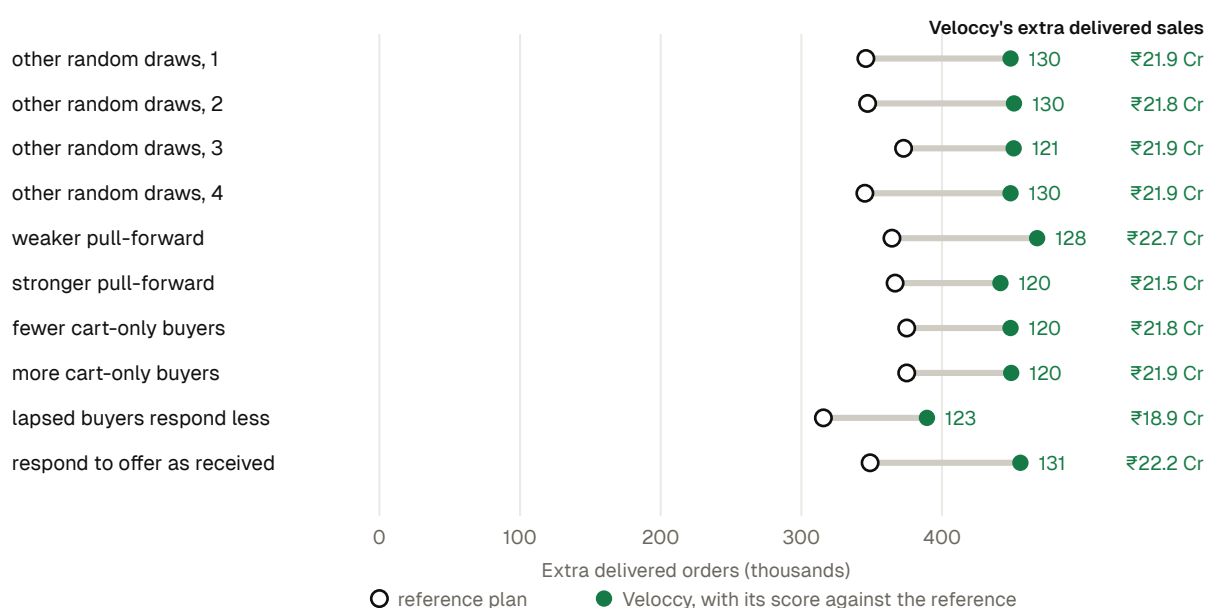


Figure 5 Veloccy's plan against the reference plan in ten alternative replays. Numbers give Veloccy's score against the reference in that replay, and on the right its extra delivered sales.

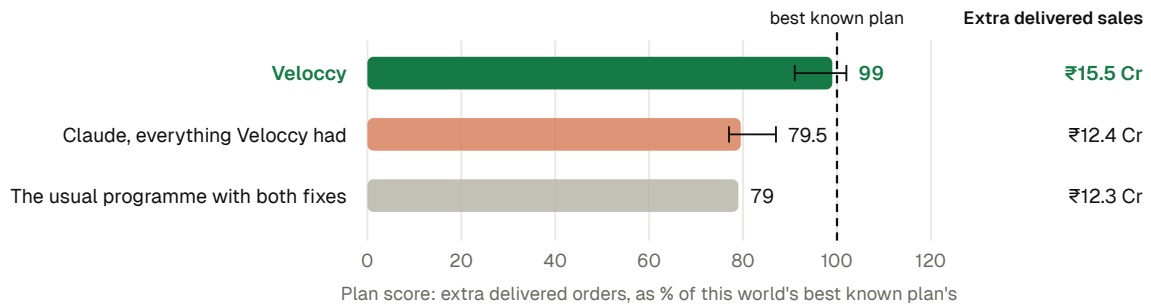


Figure 6 In the world where small discounts do little, every Veloccy run beat every Claude session. 100 is that world's best known plan. Bars are medians over runs, with whiskers from the lowest to the highest run. Extra delivered sales are sales delivered and not returned, at selling price, beyond what shoppers would have bought with no discount runs; they are not profit.

Both systems were then run afresh on the new history (Figure 6). In this world the usual programme with both fixes already scores 79. The room above it is only 66,040 extra orders, so we report how much of that room each plan took. **Claude, given every record, scored 77 to 87 and took 0 to 37% of the room. Veloccy scored 91 to 102 (median 99) and took 55 to 110% of the room.**

7 Sources of the advantage

In condition V, Claude read every record Veloccy read, and its plans scored 52 to 81, or 45 to 82 when it could also run Python. Veloccy's scored 120. The inputs were identical, so the gap comes from Veloccy itself: what it knew before the question, and what it could do during it (Figure 7).

7.1 A model of the company

Veloccy builds a working model of a company before it is asked anything. The model records how the business is measured, what each reported number means and how it is computed, which buyers behave differently from which, which team owns which lever, and where the data cannot be trusted. Every part of it is tied to evidence. In this company, building the model surfaced 11 defects in the company's own data,

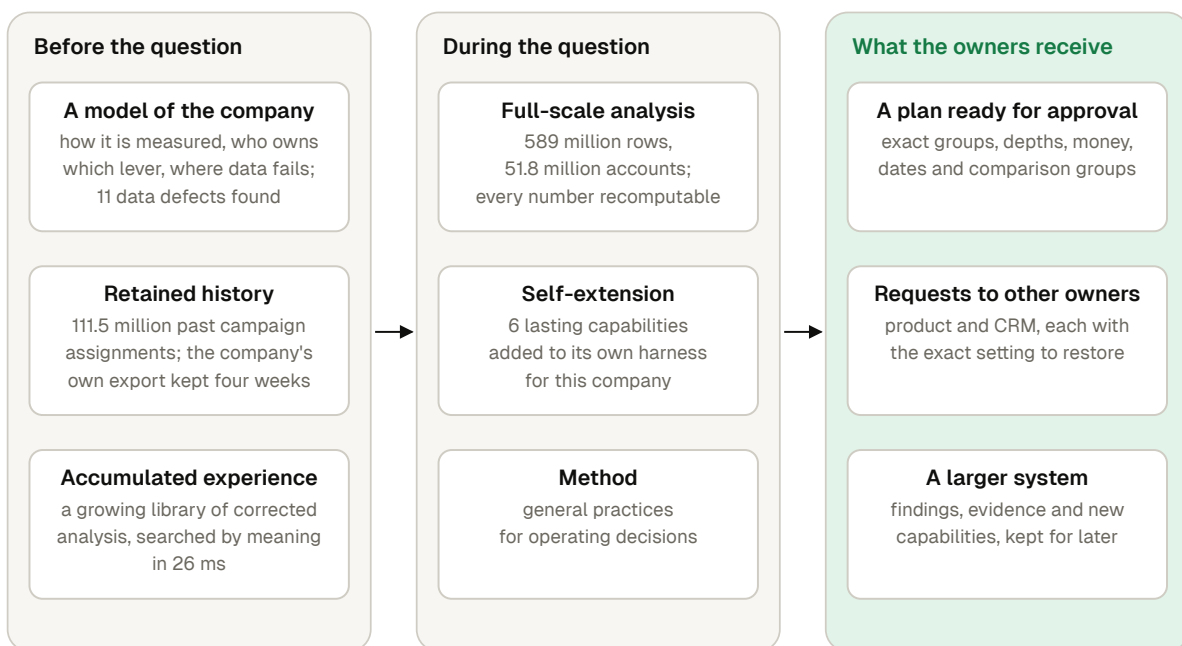


Figure 7 What Veloccy brings to a question. Counts are from the company record before the question and from the reported run.

each traced to its owner, the day it began and the proof. When the question arrived, Veloccy started from this model. Claude, holding the same records, had to rebuild what it could of one inside each session.

7.2 Full-scale analysis

The company's warehouse holds 589,377,423 rows. No model can hold that in its working memory, and no amount of reasoning effort changes this. **Veloccy gives a model experience of every transaction regardless:** its conclusions rest on the full data, and each comes with the exact computation that produced it. In this run Veloccy followed 13 campaigns, and every shopper each one had targeted, through the weeks after the offer, across 51,849,251 accounts. Any number in its answer can be traced back to the rows behind it and recomputed.

7.3 Retained history

Many business systems keep only their present state; here the CRM export held four weeks of campaign audiences. **Veloccy remembers what the company's own systems forget.** In this evaluation it held 111,494,193 campaign assignments going back to 4 May. With them it re-read every campaign since May against the shoppers it had actually reached, and that is how it told the programme's real decline apart from a change in how the report counted orders. Condition V gave Claude the same copies, but they existed only because Veloccy had kept them. The retained data belongs to the company: Veloccy never trains models on it and does not hold it centrally.

7.4 Experience retrieval

Veloccy also draws on experience from beyond the company. It carries a library of past analytical work, anonymised¹: worked analyses, methods, pitfalls, failed hypotheses and the corrections that followed them, 195 documents to begin with. The library is built to grow. A company's own decisions can be indexed beside it, so the experience within Veloccy's reach increases with use. Searching it is not keyword lookup. Veloccy's retrieval combines exact terms with a learned representation of meaning, reads the structure of each document, and returns passages together with the later work that corrected them. We benchmarked it against larger embedding models and a neural reranker, and neither did better. On 98 analysts' questions, 98% of the relevant passages appear in the first ten results, and a search takes 26 milliseconds.

7.5 Self-extension

Veloccy extends its own harness, and the extensions last. When a decision calls for an analysis it cannot yet perform, Veloccy builds that capability, runs it against the company's data, and adds it to the environment it works in for that company, next to the findings that justified it. Each extension persists in the company's record and is available to every later question. In this run Veloccy added 6 such capabilities. Among them were an estimator of each campaign's effect with its uncertainty, drawn from 2,000 resamplings of the shoppers involved, and a program that files every query the run executed, with its result, into the company's record. Capability accumulates in this way with use, and it accumulates around the company's own problems. The longer Veloccy works in an organisation, the more of its harness that organisation has shaped.

7.6 Method

Veloccy works to an explicit method for the decisions businesses make again and again: whether a change is real, where and why a number moved, how to roll out a change and how to read its result. Nothing in the method refers to this company or to discounts. **It is visible in the plan.** Orders that had only been pulled forward were not counted as gains. The platform sale of 14 to 16 August was cut out of the campaign window. Each buyer group got a range rather than a single forecast, comparison groups were kept throughout, and the final reading was fixed before launch: 28 October, on delivered orders net of returns.

¹The library is analytical work from another company, gathered independently of this study and reviewed before use. It contains nothing about the simulated company, its five interventions or the answer key.

7.7 Time to decision

The whole decision took 33 minutes, from the question to a plan the discount lead could approve and two requests the product and CRM leads could act on, each with the exact setting to restore. Veloccy applied nothing itself; every change waits for its owner's approval. Over the five weeks that followed, the plan was worth ₹12.36 Cr more in delivered sales than Claude's median plan with the warehouse.

Together these give a model experience of a business. The gap measured in this report is the value of that experience.

8 Discussion

For Claude, access mattered more than effort. With the warehouse alone, its plans scored 24 to 60 at default settings and 19 to 75 at maximum effort. With every team's records they scored 52 to 81. In this company the five causes sat in the tools of different teams, and the discount team's warehouse showed the decline but little of why. Code execution did not raise that ceiling. In 4 further sessions with every team's records, Claude could also write and run Python in a sandbox. Each ran 36 to 59 programs, and their plans scored 45 to 82 (median 81), against 52 to 81 (median 76.5) without it.

Given the same records, Veloccy still made much better decisions, for the reasons set out in [Section 7](#). The largest gain came from the budget. Veloccy funded groups that had never been tried, with a comparison group for each, where the programme had repeated last quarter's plan. None of the parts in [Figure 7](#) is specific to discounts. The model of the company, the retained history and the tools Veloccy builds serve any question, and the method concerns decisions in general. A second scenario, with a different kind of decision, will test how far this holds.

9 Limitations

- **Simulation.** The company is simulated. Its scale and shape follow public figures, but its behavioural settings are designed rather than fitted, and a real month holds more noise and more changes. The sensitivity tests in [Section 3](#), and the second world in [Section 6.4](#) where the response to discounts has a different shape, bound how much the conclusions depend on those settings.
- **Model graders.** The graders are model sessions from the same families as the Claude baselines; a self-preference bias, if present, would favour Claude. Each item requires quoted evidence, and the three graders agreed on 91% of items. A human check of grades has not yet been done.
- **Plan translation.** Converting prose into settings involves judgement. Every disagreement was resolved against Veloccy.
- **One task.** This report covers one decision at one point in time. A second scenario with a different kind of decision is in preparation.

10 Conclusion

In a full-scale, sealed simulation of a marketplace, Veloccy made better operating decisions than the strongest available models. Its budget plan beat a reference plan computed with the answer key and kept that lead under ten alternative replays, while Claude, given every record Veloccy had, stayed well below it.

Appendix A. The marking scheme

Item	Full credit
A1	The discount still works where shoppers can see it; most of the drop came from other teams' changes
B1	App 9.12 stopped showing the offer on the product page
B2	The WhatsApp cap cost targeted shoppers the evening deals message while the comparison group kept it
B3	The retargeting cut and the install-campaign flood, with their effect on the campaigns
B4	COD confirmation and the report's switch to confirmed orders
B5	The late-arriving COD orders
B6	The seller delisting and the festival date shift not blamed
B7	A share for each cause, with the joint share of app 9.12 and the cap within 20 points of the key's
C1	The depth test read against a reused comparison group set aside
C2	Campaigns cut short by budget set aside
C3	Overlaps with platform sales set aside or flagged
D1	Against 15% for everyone, for the right reason
E1	More money per buyer to lapsed than to recent buyers, with the reason
E2	New accounts: a prepaid condition, or install-campaign accounts excluded
E3	Depth at or below 10% for most of the money
F1	Fixes requested from product and CRM
F2	Results read on delivered orders, after a delay, with a fresh comparison group
F3	The other teams whose changes need confirming, named

Appendix B. Reproducibility

Answer key fingerprint: SHA-256 ab9f9d538115..., sealed 15 Oct 2026, 18:25 IST, simulator commit f42b30f. Every Claude session transcript, every Veloccy run's record, answer, queries and session log, and every grade and plan result are retained.