ORDERED BY THE NAME, PLACED BY THE BASELINE: WHAT A PERSONA INSTRUCTION CAN AND CANNOT SPECIFY

Anonymous authors
Paper under double-blind review

ABSTRACT

A persona instruction is a common interface for delegating judgment to a language model, and it works by naming a practitioner. We ask what the naming retrieves and whether it can place the policy at a chosen practitioner's level, in a domain with a public record of practice and a counterfactual value for every option: fourth-down decisions in professional American football, against twelve season cohorts whose decision thresholds span 3.65 points of win probability. Displacements below are in the same threshold units unless marked.

In both frontier models the displacement is ordered by how closely the name names the domain. With the modifier held fixed, instructing a model to decide as a *typical zorbil*, a *typical baker* or a *typical NFL head coach* displaces its policy by 1.12, 3.35 and 4.94 points — three ordered levels, separated in that model's own intervals and reproduced in a second model in a paired test. On a synthetic discipline with the same decision structure and no public record the displacement does not survive in either of the two models with an arm in all four cells — zero in one, reversed in the other — including one that identifies the structure as football on 30 of 30 probes.

Both models track the five named targets, every configuration is shown to miss at least one of them, and which instruction lands closer reverses between them. Naming each of five cohorts spanning the recorded range gives a fitted slope on the measured threshold of 0.32 $[0.09, 0.58]$ in one model — a third of the correct scale — and 0.97 $[0.50, 1.56]$ in the other; both exclude zero, only the first excludes one, and only the first recovers the five targets' exact rank order. Landing is a separate outcome and it divides the models: showing a cohort's decisions beats describing them by $2.579\,pp$ of mean absolute error $[1.534, 3.214]$ in the first model and loses by $2.754\,pp$ $[1.305, 3.958]$ in the second. Three of five rungs fall inside their target's interval in one condition and two in another, none all five, and where they land follows the difference in the two models' optimizer-prompt baselines.

The activation direction we extract acts as a signed shift in response rate, and installs no value dependence the gate can report. A third model, open-weight, passes that gate under neither the optimizer nor the persona instruction, so it is measured on the go-rate axis. There the order reverses at the step it rests on: an unrelated occupation displaces its policy *further* than the domain's own. The ordering is not invariant across the models tested. A coefficient on the mean activation difference between its two prompts moves that rate over 63.7 points of the cohorts' 22.8 while the rate where going is clearly worse tracks the rate where it is better at $\rho = +0.98$. Agreement with the persona prompt rises, but directions carrying no persona contrast do the same once swept as far, and so do three from other role prompts: at the real direction's peak its excess over a rate-matched control is $+0.034$ $[-0.035, +0.108]$. Applied backwards it moves decisions *away* from the persona where a rate-matched control moves them toward it, -0.471 $[-0.569, -0.369]$ — signed, beyond this control, rather than proportional.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Ask a language model to decide as a veteran practitioner would and it will. The persona instruction is a common interface for delegating judgment to these systems (Zheng et al., 2024), and it works by naming somebody and expecting the model to emulate their judgments. The naming does not deliver the practice: asked to decide like an NFL head coach on fourth down, the frontier models of Table 3 adopt policies more conservative than any season cohort from 2014 to 2025, and the conservatism belongs to no era in particular. That much is background, established before this paper and re-measured here. This paper asks the three questions it leaves behind.

What is retrieved? The instruction is a string with parts: a role, a role that exists, and modifiers. In both frontier models the displacement is ordered by how closely the name names the domain: a nonsense token moves the policy 23% of the way to what the domain's own occupation reaches, and a real occupation unrelated to the domain two thirds of the way. Dropping the one adjective and getting the occupation wrong cost similar amounts in one model, with neither shown to be the larger; in the other the adjective costs more and that difference is established ($+1.61\,pp$ $[+0.53, +2.94]$, $p = 0.004$, Table 8). The order is not invariant across models: in the open-weight model of Section 5 the unrelated occupation displaces more than the coach (Appendix J.1).

What does the effect require? On a decision problem with the same structure and no public record attached to it the effect almost entirely disappears, even in a model that recognises the structure as football on every probe.

Can the move be directed? We name five cohorts spanning the recorded range and show each sixteen of its own decisions. Both models move conservative by about five points under the shown rungs at the mean target — the per-cohort shifts run wider than that, and in one model wider than the cohort range itself — and the difference in where they land follows the difference in their own optimizer-prompt baselines. Both track the named cohort with a positive slope, one at a third of the correct scale and one at a scale indistinguishable from correct; only one recovers the cohorts' rank order, and each of the four settings is shown to miss at least one target, two of them all five. A coefficient on one direction extracted from residual activations moves the go rate across part of the cohort range and leaves the policy without the dependence a threshold requires. Norm-matched control directions carrying no persona contrast reproduce what agreement it buys, once they are swept far enough to move the go rate as far; what the contrast carries is the sign. **Movement, target discrimination and target placement come apart: an instruction can displace a policy reproducibly, track the targets at a fraction of their spacing, and still miss their levels.**

Relation to our earlier study. The temporal framing, the ruler and the persona threshold come from an earlier study of ours, neither published nor under review. Nothing here depends on it: Section 2 rebuilds the ruler from public play-by-play and every threshold quoted is re-measured, with that run's items, prompts and figures in the artefact.

Status of each section. Everything before Section 4 is **exploratory**: the instrument and its decision rules were developed while analysing the data they are applied to, including two claims the data withdrew. Section 4's hypotheses were **preregistered**, its design amended from two targets to five after a first collection (Appendix E); it carries the paper's main claim, and its central registered hypothesis failed.

2 INSTRUMENT

2.1 THE RULER

A policy is only extreme relative to something. Ours is the recorded practice itself: every fourth-down decision in the public play-by-play record (Carl and Baldwin, 2026) from 2014 to 2025, with the counterfactual value of each option attached (Baldwin, 2026) — $46{,}673$ of the $47{,}273$ recorded (98.7%), between 98.3% and 99.3% in every season (Appendix A). The record reaches back to 1999; the window starts at 2014 because coverage before it is 29.8%.

We summarise a policy by its **threshold** τ — the decision margin at which a fitted logistic $\Pr[\text{go} \mid m]$ crosses 50%, expressed in win-probability points forgone at indifference. The margin m, called

`go_boost` throughout, is the win probability gained by going rather than by the better alternative, so positive m means going is the higher-value option. A calibrated optimizer has $\tau = 0$; conservatism — throughout, relative to that value model — is positive. Season cohorts from 2014 to 2025 — each an aggregate over every team and coach in that season, not one practitioner — decline broadly across the window, with cluster-bootstrap intervals resampled by game.

The fit depends on which situations it is fitted over. On each season's natural distribution, overwhelmingly clear punts, cohorts run from $5.26\,pp$ to $2.15\,pp$, reproducing the published ruler. Reweighted so the five margin bands carry equal mass, matching the 300-item evaluation set, they run from $6.10\,pp$ to $2.45\,pp$; every cohort moves 0.30 to $0.91\,pp$ more conservative and the rank order is nearly but not quite preserved: Spearman $\rho = 0.965$, the eight least conservative seasons holding their places while the four most conservative reorder among themselves — 2016 takes first place from 2014, which falls to third, so the range's upper end changes hands as well as value. We use the reweighted thresholds, because a model's threshold is comparable with a cohort's only if both are fitted over the same situations.

Ruler position $P = (\tau - 2.45)/3.65$, on the reweighted bounds, places any policy on that span: 0 is the boldest of the twelve cohorts, 1 the most cautious, above 1 outside their range.

2.2 R2: WHEN A THRESHOLD MEANS ANYTHING

`go_boost` correlates 0.66 with yards-to-go in this record, so a policy running "short yardage, go" produces a strong univariate correlation with the value model and a tight-looking threshold while having no relation to win probability. One model in our set does exactly this (Appendix F). We therefore report τ, and quantities derived from it, only where a logistic regression of the decision on standardised {`go_boost`, `ydstogo`, `yardline_100`, `score_differential`, `qtr`}, with errors clustered by item, gives a positive `go_boost` coefficient at $p < 0.01$. An arm failing R2 has **no reportable threshold, not even a null one**, and is reported in go-rate units only. R2 is a gate, not a finding: it is introduced here and applied to the study whose instrument it tests, and every arm reported under it passes. The cohorts pass it too, as a ruler used to judge model thresholds should: refitted on the record with errors clustered by game, every season in the window carries an independent positive `go_boost` coefficient, $+0.87$ to $+1.64$ at $p < 10^{-18}$ (Appendix A).

2.3 ITEMS, ARMS AND QUALITY FLOORS

Evaluation uses a frozen set of **300 situations**, stratified across five margin bands with a fixed seed and rendered as anonymised structured text with three options and a letter-then-one-sentence response format. Option order is shuffled per item and the mapping recorded, so a position bias does not read as conservatism. Each arm pairs a persona condition with an **optimizer condition measured in the same run**, since baselines differ between models and domains and borrowing one across runs would confound the effect with them.

Three floors void an arm, all fixed before the runs they govern. **Parse rate** must reach 0.90; one frontier arm failed it silently (Appendix D). **Margin sensitivity**, the raw go-rate difference between extreme strata, must reach $15\,pp$, below which τ divides by a slope near zero. **Domain competence**, a floor against obvious nonsense rather than a test of football knowledge, voids an arm taking a field goal from beyond 65 yards on more than 10% of items. Every result below carries its parse rate and sensitivity; Appendix D has the condition strings.

Reproduction. The earlier run measured the persona threshold at $6.83\,pp$ $[5.73, 8.34]$ on claude-sonnet-4-6; this harness on claude-sonnet-5 gives $\mathbf{5.64\,pp}$ $[4.69, 6.99]$. The intervals overlap, and that is a check on the harness rather than a replication: the two differ in model generation, in harness, and in the instruction — the earlier persona assigns the role in its own sentence and asks for the habits of the job outright where this one is the bare clause (Appendix D). That arm also rejects `temperature` and ran at the provider default, so it is not temperature-matched to the original's 0.7.

Arm	Instruction	τ	95% CI	Sens.
P0	optimizer	-0.89	$[-1.47, -0.33]$	$83.3\,pp$
A7	a typical **zorbil**	$+0.23$	$[-0.24, 0.73]$	$86.7\,pp$
A6	a typical **baker**	$+2.46$	$[1.87, 3.21]$	$84.2\,pp$
A8	a typical **NFL head coach**	$+4.05$	$[3.29, 5.04]$	$73.3\,pp$
A1	an NFL head coach	$+2.30$	$[1.65, 3.07]$	$81.7\,pp$

Table 1: gpt-4.1, run OC: 300 items, 2 repetitions, 100% parse. All arms pass R2 ($\beta = +1.76$ to $+2.78$, $p < 10^{-3}$). *Zorbil* is not a word. A1 is not part of the three-noun comparison — it is A8 without *typical*, included so the modifier is readable within the run. The second model's arms, and the paired bootstrap of every step in both, are in Appendix G.

3 What the instruction retrieves

The persona this paper ablates is the core clause of the earlier study's registered instruction, with its role assignment and its explicit request for the habits of the job removed (Appendix D): *decide exactly as a typical veteran NFL head coach would*. It carries an instruction to adopt a role, a role that exists, and two modifiers, stripped one at a time in a single grammatical frame so that sentence form is not itself a variable.

The name is not an irrelevance, and it acts by degrees. Three instructions differing only in the noun, in one run against one optimizer baseline:

The total displacement of $4.94\,pp$ divides into three ordered steps, each between arms whose intervals do not overlap: **naming anybody at all** ($+1.12\,pp$, 23%), **making the name a real occupation** ($+2.24\,pp$, 45%), and **making it the domain's occupation** ($+1.59\,pp$, 32%). They are not equal — the middle is twice the first and 1.4 times the third — and it is the ordering rather than the spacing that the run establishes.

Paired within bootstrap draws the smallest step is $+1.12\,pp$ $[0.65, 1.64]$; all six ladder steps across the two models are positive, five at $p < 0.001$ and gpt-luna's last at $p = 0.005$, uncorrected — under Holm over the ten paired occupation tests nine stay below 0.05, the exception being the contrast that is not significant uncorrected either (Appendix G).

A second model reproduces the ladder from a different starting point. gpt-luna's optimizer sits $3.92\,pp$ conservative of gpt-4.1's, and the same three nouns displace it by $+1.18$, $+3.25$ and $+5.15$ — within $0.21\,pp$ of gpt-4.1's $+1.12$, $+3.35$ and $+4.94$ at every rung, middle step largest in both, every step clearing zero paired (Appendix G). Their coach arms land $4.13\,pp$ apart, which is Section 4's result appearing here: the instruction moves the policy a fixed distance and the baseline decides where it ends. In gpt-luna it ends outside the recorded range.

What the movement costs does not carry over. In gpt-4.1 only the domain's own occupation costs sensitivity, 83.3 to $73.3\,pp$; in gpt-luna it falls at every rung, by 5.8, 23.3 and $34.2\,pp$ (Appendix G). The displacement replicates and its price does not, so the cost is a property of a model rather than of the instruction.

One modifier does most of the rest. Isolating *typical* gives $+0.87$ and $+0.88$ on two models whose baselines differ by more than a factor of two; *veteran* gives $+0.28$ and -0.20, a sign we do not report as an effect. Replacing *typical* with *aggressive* moves the threshold $-12.94\,pp$ from the same arm that *typical* moves $+1.95\,pp$ from, landing at $P = -3.61$ against *typical*'s $+0.47$: the modifier names a position among practitioners and the two words name opposite ones (Appendix G).

Set against each other, the adjective and the occupation give similar estimates: *a typical baker* and *an NFL head coach* differ by $+0.16\,pp$ $[-0.24, +0.58]$, centred near zero; we set no equivalence margin in advance and claim none. (Units above: the first pair is in go-rate ruler widths, the rest threshold points against that run's A1 arm, P ruler position.)

3.1 The effect requires the domain, not the structure

If a role instruction were a general prior toward caution it would work anywhere. We ran the same instructions on a synthetic discipline with the same decision structure — a risky action, a safe one

conceding position, a small certain gain — the same margins carried over item by item, and no practitioner anyone has written about.

In football every naming produces a large displacement: $+2.48$ and $+2.56$ ruler widths for the real coach on claude-sonnet-5 and gpt-4.1, $+1.34$ and $+1.55$ for *a typical veteran crew director* — the *invented* role of Appendix G's A4 arm with the two modifiers restored, not the unrelated occupation, which is A6's baker. In the synthetic domain none does, the real coach included: it moves claude-sonnet-5 by -0.85 widths and gpt-4.1 by $+0.00$, and the invented name moves claude-sonnet-5 by -0.45 and gpt-4.1 by $+0.09$. **The conservative displacement requires the domain, and the name scaled it.** Neither of the two models with arms in all four cells retains a football-sized conservative shift: as a share of the same model's football effect under the same name, gpt-4.1's is 0% and $+6\%$, claude-sonnet-5's -34% and -34% — **one all but vanishes outside the domain and the other reverses at a third of the magnitude**. The $+6\%$ is a conservative residual, and a small one. Nor is gpt-4.1's zero a ceiling effect: the football effect is a *fall* in the go rate, its 96.6% synthetic baseline leaves room for one, and the invented name does produce one ($-2.1\,pp$; Appendix G). A third model has only the football real-name and synthetic invented-name arms, so its residual crosses the name condition as well as the domain and is not a domain residual (Appendix G).

Asked what the synthetic rules resemble, one model named American football on **30 of 30** probes, mapping the three actions onto go, punt and field goal explicitly, and showed no persona effect on those same rules. The other named no sport on any probe and likewise showed none. **Recognising the structure is not sufficient in this control.**

The synthetic margin was carried from the football items to match the difficulty distribution, so it is a stratum label rather than the value of pressing under the synthetic rules. Thresholds and sensitivity are therefore not interpretable on those arms and only the go-rate displacement is reported; one model also spends 69% of its synthetic optimizer choices on a single action, which reflects the payoff structure we wrote.

4 CAN THE INSTRUCTION SPECIFY A COHORT?

The hypotheses tested here were registered before any ladder data were collected; the ladder itself was redesigned afterwards, from two targets to five. Appendix E gives registered, executed and reason for each element. Everything before this section was not registered at all.

Everything above shows that a persona instruction **moves** the policy. Nothing shows that the move can be **directed at a chosen cohort**, the question this instrument was built to leave open: whether the policy can be *steered to a chosen point between the baselines*.

Five cohorts spread across the ruler: 2016 at $\tau = 6.10\,pp$, 2017 at 5.60, 2018 at 4.31, 2022 at 3.45, 2025 at 2.45 — a range of $3.65\,pp$, the full span of the twelve cohorts, with the most even spacing five seasons allow. Each is named two ways: **described** (P4: "goes for it about $X\%$ of the time when going is the higher-value option") and **shown** (P5: "decisions actually made by NFL head coaches in the *year* season", then sixteen of them in the evaluation's format). **The shown rung therefore names the cohort as well as displaying it**, and the described rung does not; nothing below separates the two channels.

Two targets determine a slope and a level exactly; five leave three residual degrees of freedom and make linearity testable. The earlier two-target run is in Appendix I. Targets use the reweighted thresholds of Section 2. The blocks reproduce each cohort's go rate within each margin band, but the sixteen delivered are the first sixteen of a block of sixty-four: **the delivered prefix spans** $12.50\,pp$ **against these cohorts' own** 21.93, 0.570 of the contrast, in three distinct rates for five cohorts. **We therefore claim no ceiling of** 1 **for the slope**, nor of 0.570, for the reason Appendix C gives.

gpt-4.1's shown rungs land inside three of the five cohorts' intervals — the arm's point estimate within the cohort's own bootstrap interval, not the weaker test of overlapping intervals (2016, 2017 and 2018, $|\mathrm{err}| = 0.21$ to 0.75; 2022 misses by 0.99 and 2025 by 1.59) — and beat doing nothing at every target. **gpt-luna's shown rungs land inside none**, overshoot by 2.0 to $5.1\,pp$, and are worse than its own optimizer at four targets of five; its *described* rungs reach two.

	optimizer	P4 described	P5 shown	R^2/sd: P4, P5
slope b, 1 is correct				
gpt-4.1	—	0.26 [0.14, 0.40]	0.32 [0.09, 0.58]	0.92/0.13, 0.93/0.15
gpt-luna	—	0.51 [0.25, 0.83]	**0.97** [0.50, 1.56]	0.76/0.50, 0.61/1.36
level at the mean target, 4.38 *pp* is correct				
gpt-4.1	−0.99	0.95 [0.37, 1.61]	**4.65** [3.84, 5.69]	
gpt-luna	3.04	**4.91** [3.95, 6.37]	7.96 [6.64, 9.64]	

Table 2: Five targets, three residual degrees of freedom. Every interval is a **paired cluster bootstrap**: the 285 source games resampled once per replicate, all five arms refitted inside that draw, and each cohort's named threshold redrawn from its own interval in the same replicate, so both sides' uncertainty is carried at once. It replaces the five-point least squares ± earlier versions reported — a different quantity, not a conservative one — and is wider for gpt-4.1, narrower for gpt-luna. Appendix I has both; Table 13 the five thresholds behind every fit.

4.1 Where the policy lands tracks the starting point

Both models respond to being shown a cohort's decisions the same way: the policy moves conservative by about five points of win probability (+5.64 *pp* for gpt-4.1, +4.92 *pp* for gpt-luna, at the mean target), and naming the occupation once in Section 3 moved the same two models +4.94 and +5.15. Figure 2 in Appendix I plots all ten arms against their targets, as a view of Tables 2 and 13, which hold every number this section reports, rather than a source. They differ in where the movement starts. gpt-4.1 begins at −0.99, bolder than any cohort, and five points of conservatism puts it at 4.65 against a correct 4.38 — close in practice, and not distinguishable from it once both sources of uncertainty are carried ([3.84, 5.69]; the ±0.07 of the five-point fit separated them on the line's scatter alone). gpt-luna begins at 3.04, inside the range already, and the same movement carries it to 7.96, past every cohort. **Neither places itself *at* the target, and the baseline shift dominates what tracking there is: gpt-4.1's shown rungs sit** 5.6 *pp* **above its own optimizer and move** 1.2 *pp* **across the cohorts'** 3.65**, so only one of the two starts inside the range.**

The movement costs differently in the two. Under the shown rungs gpt-4.1 retains 71–85% of its optimizer's 81.7 *pp* of margin sensitivity; gpt-luna retains as little as a third of 75.8 *pp*, and its least sensitive arm (26.7 *pp*, the 2017 target) is also its most displaced — its residual sd of 1.36 *pp* against 0.15 follows from the same flattening.

The slopes run the other way from the levels, and **which line is better determined depends on the question**. Asked whether five points could have come from a flat line, gpt-4.1's slope separates from zero ($t = 6.4$, $p = 0.008$) and gpt-luna's does not ($t = 2.2$, $p = 0.12$), its scatter being nine times larger. Asked whether the fit would stay positive if the 285 source games were drawn again — Table 2's bootstrap, on the arms' 2,988 to 3,000 rows rather than five summaries — **both do**, with no replicate of 4,000 reaching zero in either.

A positive trend is not a recovered ordering, and the two come apart here. On an exact permutation test gpt-4.1 recovers the cohorts' rank order exactly on both rungs ($\rho = +1$, $p = 0.008$) and gpt-luna does not (+0.70, $p = 0.12$ shown; +0.90, $p = 0.042$ described), its 2017 arm landing above its 2016 one. So: **both models track the named cohort with a positive slope, one at about a third of the cohorts' scale and one at a scale indistinguishable from correct; one also orders them correctly; and each of the four is shown to miss at least one named target.** Appendix I tabulates the six quantities this paragraph separates.

4.2 The registered hypotheses

H6a, every rung closer to its target than the optimizer. Holds throughout for gpt-4.1; fails for gpt-luna at four of five targets on the shown rungs, partly structurally, since its baseline sits inside the target range and a persona that moves conservative cannot approach the bolder cohorts from there. The hypothesis is only meaningful for a baseline outside the range on the correct side, which the registration did not say.

H6b, showing beats describing. Mixed: it holds in gpt-4.1 on level and landing error, is not ordered on the fitted value association in either model (Table 15), and **is not established on slope in either model**: differencing the two fits over the same five targets gives $+0.06 \pm 0.09$ and $+0.45 \pm 0.49$, $t_3 = 0.6$ and 0.9 against a critical 3.18, so the ordering of the point estimates is all the slope supports. In gpt-luna it reverses on landing error. On mean absolute error the settings pair off across the models rather than ordering by rung (Appendix I), so the rung predicts the error within a model and not between them. **Each model has exactly one rung whose level is not distinguishable from the correct** 4.38 — gpt-4.1's shown and gpt-luna's described — while the other separates in each (Table 2). A crossing point is not a placement: gpt-luna's described arm still misses individual targets by up to $1.28\,pp$ at slope 0.51, and it is the one arm in the paper the R2 gate can remove (Appendix I).

H6c, the targets are distinguished. On the five-point fit it holds in gpt-4.1 ($p = 0.008$ shown, 0.007 described) and not in gpt-luna ($p = 0.12$ and 0.058); **on the paired bootstrap it holds in both**, all four intervals excluding zero, and we report the disagreement rather than the more favourable half. Registered as a separation threshold on two targets, which the five supersede; the registered form passes in gpt-luna and at two of three k in gpt-4.1 (Appendix I). We report the five-target form because it has residual degrees of freedom, and on it **the targets are distinguished in both models and ordered correctly in one**, the distinction H6c as registered does not draw.

H6d, error non-increasing in k. Not evaluable — the design fixes $k = 16$ — and not monotone in either model where k varied (Appendix I).

4.3 The prompts respond to the cohort they name

One comparison needs no model internals. The described rung states the go rate of the full sixty-four-example block, a $21.9\,pp$ contrast across the cohorts; the shown rung displays the first sixteen, a $12.5\,pp$ one. The rung carrying less of the contrast has the larger slope in both models (0.32 against 0.26, 0.97 against 0.51) — but refitting both rungs inside one game draw puts the difference at $+0.06$ $[-0.20, +0.32]$ in gpt-4.1 and $+0.45$ $[+0.00, +1.05]$ in gpt-luna, so **the rungs' slope ordering is a pattern in the point estimates that one model's interval separates from zero and only barely** (Appendix I). What those draws do separate is the level, by 3.1 to $3.7\,pp$ in both.

The rest of this subsection is exploratory: the quantity sits outside R2's gate, and in gpt-4.1 no single arm survives a Holm correction over the ten tests of the five-cohort run. The obvious reading of a null, that the model ignored the examples, is wrong twice over. The slopes are direct evidence that the target is read, and in gpt-4.1 the multivariable fits show what reading it does: `go_boost` rises from $\beta = +1.46$ in the optimizer arm to $+2.25$ and $+2.29$ under P4 and $+2.47$ to $+3.24$ under P5, while the yards-to-go dependence the optimizer carries (-0.93, $p = 0.0008$) loses significance in seven of the eight persona arms and reverses in the eighth. **gpt-luna goes the other way on both counts** in both ladder runs, and here one of its arms fails R2 outright (run L2, Appendix I). **The cohort-conditioned prompt changes the fitted association with the value variable. It does not make the model the practitioner the examples came from.**

Margin sensitivity, the raw go-rate gap between extreme strata, *falls* where that coefficient rises, and neither half is evidence alone. We therefore report the **average partial effect**, in probability units and comparable across base rates (Appendix I defines it and says why neither half suffices). In gpt-4.1 it rises in all eight persona arms of run L2 by $+7.34\,pp$ on average $[+1.91, +13.91]$ and by $+5.78$ over the five-cohort run's ten, and **gpt-luna reverses it in both**, all ten point estimates falling (Appendix I).

5 Discussion

5.1 What the persona is doing instead

What follows is interpretation rather than measurement; the three results below constrain it without selecting it. Displacement precedes competence (Appendix B), structure alone does not produce it and the domain must (Section 3.1), and the name reaches the domain by degrees — nobody, a real occupation, the domain's own — while one adjective is worth as much as the occupation being the right one (Section 3).

Together they suggest the instruction is not retrieving a practitioner but selecting a *register*: a way of deciding that the corpus associates with the domain, keyed by a request for its conventional form. There is no baker's fourth-down judgment to retrieve, so what is retrieved under *baker* is the coach's, labelled with the name the instruction supplied — but only two thirds of it. The name is not the content; it is how much of the domain's register the instruction reaches. *typical* asks for a central value and *aggressive* for a tail, and the model moves the opposite way under the second, further than the record extends.

The account is about the level, not the discrimination. Both models separate the cohorts when shown their decisions, so both extract something that differs between the targets, which a register with no years in it cannot explain; what it does explain is the floor they cannot get below. The present data cannot select among accounts, and we say so rather than choosing one.

5.2 What a representation-level intervention reaches

Section 4 shows the prompt does not select which cohort the policy lands on, which does not mean no other intervention can. Two constraints narrow what the attempt can be compared with. Every model whose threshold passes R2 under the optimizer instruction is API-only and every open-weight model with activation access fails it there, across the one series tested (Appendix B), so the quantity is the **go rate when going is the higher-value option**, the axis the cohorts are also summarised on; and a sweep is affordable only from the answer-token logits rather than by generating an answer. The two read-outs put this model's unsteered policy at nearly the same place (76.4% generating against 77.4% at the answer token, same instruction), but neither rate is comparable with Section 4's, which is a threshold on other models (Appendix J).

The method is standard activation addition from contrastive prompt pairs (Turner et al., 2023; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024), on Qwen2.5-14B-Instruct in bf16 at layer 30 of 48 (Appendix J).

The coefficient moves the go rate across most of the cohort range. Pooled over all 300 items, each fold evaluated with the direction from the other half, it falls monotonically from 89.7% at -0.90 to 26.0% at $+0.60$ — eight of ten non-degenerate settings — and turns back outside that run, a span of $63.7\,pp$ against the cohorts' $22.8\,pp$. The direction is persona minus optimizer added with a positive coefficient, so the run has to *fall*: the persona prompt lowers this rate by $25.3\,pp$, and it does ($62.3\,pp$ sampling the same answer token on the same activations). Four of the twelve cohorts are matched within $2\,pp$, none at a setting whose policy reads the margin, and the run stops short of the lowest three (Appendix J).

No tested coefficient produced an R2-valid decision rule. R2 fails at all twelve settings under both read-outs, and there is none for a coefficient to move that the gate can report to begin with: pooled, the unsteered `go_boost` coefficient is -0.019 ($p = 0.92$). Sensitivity is negative at every setting, no interval reaches the $15\,pp$ floor, and it falls further as the coefficient rises ($\rho = -0.80$, $p = 0.002$; -0.90 sampled). The go rate where going is clearly worse and the rate where it is better move together at $\rho = +0.98$, which is what scaling a response bias produces.

Agreement with the persona prompt rises, and a vector carrying no persona contrast does the same. Item-level agreement climbs monotonically over ten settings from $\kappa = +0.426$ unsteered to $+0.605$ at $+0.20$, a paired $+0.179$ $[+0.097, +0.260]$. But a policy whose rate has moved toward the persona's agrees with it more for that reason alone, and four norm-matched controls buy $\Delta\kappa$ the same way, by where the moved rate lands rather than how far it went; against the nearest the real direction's excess is $+0.034$ $[-0.035, +0.108]$. Three further directions, extracted from role-conditioned prompts that differ from *each other* in one noun — not from the source, which carries *veteran* as well — also match it once the rate is held. **The agreement gain is not shown to be specific to the contrast** (Appendices J and J.1).

What the contrast does carry is a sign. Applied backwards the direction moves decisions *away* from the persona: at -1.00, where the rate has fallen only $4.8\,pp$ and a rate-matched control gains $+0.067$, the real direction loses -0.404, a paired -0.471 $[-0.569, -0.369]$ — an order of magnitude larger than the excess at the peak — a signed asymmetry beyond this matched control, which is not a claim about every rate displacement.

At five of the seven layers scored, adding the direction moves the policy *away* from the persona, so a layer chosen on the width of its excursion selects against the intervention being tested. **The direction is cross-fitted and the layer is not**, so a pooled number is held out with respect to the direction and inherits one layer choice. Nor can the coefficient be set: chosen on one half and evaluated on the other it leaves landing errors of 14.7 to 34.7 *pp* (Appendix J).

Whether representation-level intervention beats prompting is contested (Banayeeanzade et al., 2025; Blas et al., 2026). **The direction this section extracts, in a model whose policy passes the threshold gate under neither instruction it is extracted from, acts as a signed shift in response rate and adds no value dependence the gate can report**, and so do the three role-derived directions of Appendix J.1.

6 LIMITATIONS

Scope. One domain, chosen because it has a public record, an accepted counterfactual model and discrete options — what makes the ruler possible and what makes it atypical. All open-weight models come from one series, two frontier models are not temperature-matched to the rest (Appendix D), and the three-noun ordering reverses on the third model, so it is not invariant across the three and three models cannot attribute which orders which way. This interface is used most often where no option-level value model exists, and there only the warning transfers, not the number.

Everything before Section 4 is exploratory, developed while analysing the data it is applied to. Two claims did not survive and are withdrawn rather than removed: that personas generally cost situational sensitivity, and that the cost tracks relocation distance, which the within-model correlation does not support. On the first, 30 of the 79 arms with an optimizer arm to divide by retain 85 to 109% of its margin sensitivity — 21 of gpt-4.1's 31 against 7 of gpt-luna's 28 — so the cost is not general, and where it appears it is not small (Appendix D).

The specification result rests on two models whose lines are differently determined. Slope 0.32 against 0.97, residual sd 0.15 against 1.36: on the five-point fit the second does not separate from zero, on the paired bootstrap both do, and we report both because a reader is entitled to the less favourable one. The two slopes are not distinguishable from each other on either (0.65 ± 0.45, $t = 1.4$), so we do not establish that the models track differently; only one orders the cohorts correctly, a separate fact; with $n = 2$ we cannot say which behaviour is typical, and the account we prefer is fitted to two points. One arm is unexplained and two are short of a full complement of rows (Appendices D, I). **The stimulus is also weaker than an earlier version of this paper claimed**: the sixteen delivered examples carry 0.570 of the cohorts' go-rate contrast, not the 0.998 of the sixty-four built, so part of the compression we measure belongs to what the model was shown and we cannot say how much (Appendix C).

The controls are narrow. A baker and one nonsense token do not establish that every occupation behaves alike, only that *crew director*, being an invented role, could not separate naming a role from naming a real occupation. That separation is the baker arm's job, and it rests on one occupation. The familiarity probe is weaker still — one model declines four fifths of it — so the claim that the slope is not recall rests on two structural arguments (Appendix H).

The steering result is one model, one direction, one domain, measured in go rate because that model fails R2 — it has no value-tracking rule to move before any instruction — under a decision rule that is not Section 4's, and at a layer chosen from seven on one half of the items and measured on the other. An earlier pass read the answer letters at a position holding 1.2% of the mass; its figures were discarded and the experiment re-run, layer selection included. Its controls bound the result rather than rescue it and were swept at non-negative coefficients only, so whether a contrast-free vector can raise the go rate as the real direction does is untested (Appendix J).

R2 gates thresholds; it does not validate the value model. Whether `go_boost` is the right account of fourth-down value is inherited from the ruler, and the cohorts' own thresholds shift when refitted over a different distribution, so the human policies are not pure functions of it either. Under perturbation the parse and sensitivity floors decide nothing and R2 does work: `A5_aggressive_role` passes at $\alpha = 0.01$ and fails at 0.001, and the largest p behind any body number is 0.0098. Appendix K sets this against a literature scoring personas on task performance and survey distributions rather than on a decision policy.

AI USE STATEMENT

A large language model was used in two of the ways the submission form asks about, and the form's categories are coarse enough that the detail belongs here.

Writing. The manuscript was drafted and revised with model assistance, including the abstract, the section structure and the wording of individual claims. Every figure quoted in the text was checked against the artefact that produced it, and the authors wrote or approved every claim.

Research execution. The analysis and figure code was written with model assistance, as were several of the diagnostics it runs: the completeness check that separates rows never written from rows that failed to parse, the bootstrap intervals on margin sensitivity, the widest-monotone-run search that replaced a rank correlation after that correlation passed a single-peaked response, and the per-item agreement test between a steered policy and the persona prompt. Model-written code also contained errors that testing caught — a flip counter that read zero while the go rate moved sixty points, a degeneracy flag that fired on every working setting — and both the tests and the corrections are in the released repository.

Research ideation. Two design decisions came from model-assisted analysis rather than from the registration: cross-fitting the two steering folds so that every item is evaluated exactly once, and recognising that the steered layer had been chosen on the same half of the items it was then measured on. That is disclosed rather than repaired: the scan ran on fold a (`phase_steer_v2_layerscan`), Section 5's span pools both halves and therefore includes the half the layer was chosen on, and Appendix J reports each fold separately so the held-out one can be read on its own.

The research questions, the instrument, the preregistration and the interpretation of every result are the authors'. The authors take full responsibility for the contents of this paper, including any part drafted with model assistance.

ETHICS STATEMENT

The study involves no human subjects and no private data. No new human responses were collected: the model arms are measured on language models and the cohort thresholds, the coverage audit and the cohorts' own R2 fits are a secondary analysis of a published record. That record is publicly published play-by-play data for professional American football games (Carl and Baldwin, 2026), aggregated to season cohorts; no individual is named in any condition string or item, and the named targets are occupational roles and season aggregates.

The main finding has a deployment consequence worth setting out directly. Persona instructions are widely used to set a model's standard of judgment, and we find that naming a cohort moves both models' thresholds in the cohorts' own direction and orders the five correctly in one model of the two, that every condition we tested is shown to miss at least one of the five named cohorts' levels, and that the landing error tracks each model's own optimizer-prompt starting point. A practitioner who selects a persona string in order to obtain a particular cohort's decision threshold should not assume the level transfers, and should measure it. We report this as a limit on the interface, not as a reason to prefer any particular alternative.

Section 5 reports an intervention on residual activations that moves the model's response rate across part of the recorded range without producing a setting that passes the paper's threshold gate. It is a negative result, and we are not reporting a working method for placing a policy at a named level. What is flagged here is the mechanism rather than the outcome: an edit of this kind changes what the model outputs while leaving the instruction — the part a downstream reader can inspect — unchanged. We release the two prompts whose activation difference defines the direction, so the measurement can be checked. The preregistration for Section 4 was deposited before that section's data were collected, and Appendix E lists every point at which the executed design differs from the registered one.

Reproducibility statement

The harness, every condition string, the frozen item sets, the ruler, the in-context examples, the registration document and the list of deviations from it are available for review at

`https://anonymous.4open.science/r/review-artifact-nb-EAA1/`

which carries no author identity. Section 4's preregistration is also deposited under a DOI that was issued before that section's data were collected; the DOI identifies the authors, so it is withheld for anonymous review and will be given in the camera-ready. The deposit timestamp is what establishes the ordering, and it is verifiable from the DOI alone once that link is public. So *what* was registered can be read now, and *when* it was deposited can be checked after review. Every point estimate in the body can be recomputed from the appendices: Appendix D gives the model snapshots and dates, Appendix E the registered-against-executed comparison, and Appendices G and I the per-arm thresholds behind every fit. The interval estimates cannot: the paired bootstrap, the average partial effects and the steering agreement statistics are computed from per-item outputs, which the appendices summarise rather than contain, so reproducing those requires the released decision files and code. Which files those are is written down rather than left to be inferred: the artifact's `MANIFEST.md` lists every run directory with its files, their sha256 prefixes and their row, item and cluster counts, and names the table or section each directory feeds. It also records the checks that index is built on — which copies of a frozen input are byte-identical to it, which runs carry a file named for one and holding something else, and which directories nothing in the paper cites.

References

Allison, P. D. 1999. Comparing Logit and Probit Coefficients Across Groups. *Sociological Methods & Research* 28(2): 186–208.

Argyle, L. P.; Busby, E. C.; Fulda, N.; Gubler, J. R.; Rytting, C.; and Wingate, D. 2023. Out of One, Many: Using Language Models to Simulate Human Samples. *Political Analysis* 31(3): 337–351. doi:10.1017/pan.2023.2.

Baldwin, B. 2026. nfl4th: Functions to Calculate Optimal Fourth Down Decisions in the National Football League. R package version 1.0.7. doi:10.32614/CRAN.package.nfl4th.

Banayeeanzade, A.; Tak, A. N.; Bahrani, F.; Bolourani, A.; Blas, L.; Ferrara, E.; Gratch, J.; and Karimireddy, S. P. 2025. Psychological Steering in LLMs: An Evaluation of Effectiveness and Trustworthiness. arXiv:2510.04484.

Bisbee, J.; Clinton, J. D.; Dorff, C.; Kenkel, B.; and Larson, J. M. 2024. Synthetic Replacements for Human Survey Data? The Perils of Large Language Models. *Political Analysis* 32(4): 401–416. doi:10.1017/pan.2024.5.

Blas, L.; Jia, R.; and Ferrara, E. 2026. Psychological Steering of Large Language Models. arXiv:2604.14463.

Brill, R. S.; Yurko, R.; and Wyner, A. J. 2025. Analytics, Have Some Humility: A Statistical View of Fourth-Down Decision Making. *The American Statistician* 79(3): 393–409.

Carl, S.; and Baldwin, B. 2026. nflfastR: Functions to Efficiently Access NFL Play by Play Data. R package version 5.2.0. doi:10.32614/CRAN.package.nflfastR.

Chang, T.; Schnabel, T.; Swaminathan, A.; and Wiens, J. 2026. A Course Correction in Steerability Evaluation: Revealing Miscalibration and Side Effects in LLMs. *Proceedings of the AAAI Conference on Artificial Intelligence* 40(44): 37259–37267. doi:10.1609/aaai.v40i44.41057.

Chen, R.; Arditi, A.; Sleight, H.; Evans, O.; and Lindsey, J. 2025. Persona Vectors: Monitoring and Controlling Character Traits in Language Models. arXiv:2507.21509.

Cheng, M.; Durmus, E.; and Jurafsky, D. 2023a. Marked Personas: Using Natural Language Prompts to Measure Stereotypes in Language Models. In *ACL 2023*. arXiv:2305.18189.

Cheng, M.; Piccardi, T.; and Yang, D. 2023b. CoMPosT: Characterizing and Evaluating Caricature in LLM Simulations. In *EMNLP 2023*, 10853–10875. doi:10.18653/v1/2023.emnlp-main.669.

Deshpande, A.; Murahari, V.; Rajpurohit, T.; Kalyan, A.; and Narasimhan, K. 2023. Toxicity in ChatGPT: Analyzing Persona-Assigned Language Models. In *Findings of EMNLP 2023*, 1236–1270.

Durmus, E.; Nguyen, K.; Liao, T.; Schiefer, N.; Askell, A.; et al. 2024. Towards Measuring the Representation of Subjective Global Opinions in Language Models. *COLM 2024*.

Gupta, S.; Shrivastava, V.; Deshpande, A.; Kalyan, A.; Clark, P.; Sabharwal, A.; and Khot, T. 2024. Bias Runs Deep: Implicit Reasoning Biases in Persona-Assigned LLMs. In *ICLR 2024*. arXiv:2311.04892.

Kim, J.; Yang, N.; and Jung, K. 2024. Persona is a Double-edged Sword: Mitigating the Negative Impact of Role-playing Prompts in Zero-shot Reasoning Tasks. arXiv:2408.08631.

Kong, A.; Zhao, S.; Chen, H.; Li, Q.; Qin, Y.; Sun, R.; Zhou, X.; Wang, E.; and Dong, X. 2024. Better Zero-Shot Reasoning with Role-Play Prompting. In *NAACL 2024*, 4099–4113.

Li, K.; Patel, O.; Viégas, F.; Pfister, H.; and Wattenberg, M. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *NeurIPS 2023*.

McIlroy-Young, R.; Sen, S.; Kleinberg, J.; and Anderson, A. 2020. Aligning Superhuman AI with Human Behavior: Chess as a Model System. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, 1677–1687. doi:10.1145/3394486.3403219.

Meister, N.; Guestrin, C.; and Hashimoto, T. 2024. Benchmarking Distributional Alignment of Large Language Models. arXiv:2411.05403.

Mood, C. 2010. Logistic Regression: Why We Cannot Do What We Think We Can Do, and What We Can Do About It. *European Sociological Review* 26(1): 67–82. doi:10.1093/esr/jcp006.

Rimsky, N.; Gabrieli, N.; Schulz, J.; Tong, M.; Hubinger, E.; and Turner, A. M. 2024. Steering Llama 2 via Contrastive Activation Addition. *ACL 2024*, 15504–15522. doi:10.18653/v1/2024.acl-long.828.

Romer, D. 2006. Do Firms Maximize? Evidence from Professional Football. *Journal of Political Economy* 114(2): 340–365. doi:10.1086/501171.

Salewski, L.; Alaniz, S.; Rio-Torto, I.; Schulz, E.; and Akata, Z. 2023. In-Context Impersonation Reveals Large Language Models' Strengths and Biases. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. arXiv:2305.14930.

Sandholtz, N.; Wu, L.; Puterman, M.; and Chan, T. C. Y. 2024. Learning Risk Preferences in Markov Decision Processes: An Application to the Fourth Down Decision in the National Football League. *Annals of Applied Statistics* 18(4): 3205–3228. arXiv:2309.00756.

Santurkar, S.; Durmus, E.; Ladhak, F.; Lee, C.; Liang, P.; and Hashimoto, T. 2023. Whose Opinions Do Language Models Reflect? *ICML 2023*, PMLR 202: 29971–30004.

Tang, Z.; Jiao, D.; McIlroy-Young, R.; Kleinberg, J.; Sen, S.; and Anderson, A. 2024. Maia-2: A Unified Model for Human-AI Alignment in Chess. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. arXiv:2409.20553.

Turner, A. M.; Thiergart, L.; Udell, D.; Leech, G.; Mini, U.; and MacDiarmid, M. 2023. Activation Addition: Steering Language Models Without Optimization. arXiv:2308.10248v1. Later versions are retitled *Steering Language Models With Activation Engineering* and add an author; v1 is pinned here because it is the version this method is named for.

Zheng, M.; Pei, J.; Logeswaran, L.; Lee, M.; and Jurgens, D. 2024. When "A Helpful Assistant" Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models. In *Findings of EMNLP 2024*, 15126–15154. arXiv:2311.10054.

Zhu, J.-Q.; Yan, H.; and Griffiths, T. L. 2025. Steering Risk Preferences in Large Language Models by Aligning Behavioral and Neural Representations. arXiv:2505.11615.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; et al. 2023. Representation Engineering: A Top-Down Approach to AI Transparency. arXiv:2310.01405.

A INSTRUMENT: THE SECOND RULER

τ is undefined for a policy whose choices do not depend on the margin — and below frontier scale, most policies do not. So the same cohorts are also summarised by **go rate where going is the higher-value option**, running from 24.6% in 2014 to 47.4% in 2025, a span of $22.8\,pp$. Position on this second ruler is oriented to match the first, so that higher always means more cautious.

The go-rate ruler is what makes Appendix B measurable at all. A model with no threshold still has a go rate, and in every model measured here the instruction still moves it.

What the ruler is computed on. The file behind it holds 107,107 fourth downs from 1999 to 2025. A row enters a threshold only if a counterfactual value is available for each option, and the availability is what sets the 2014 start: 17,833 of the 59,834 rows before 2014 qualify (29.8%) against 46,673 of the 47,273 from 2014 on (98.7%). Inside the window the 600 excluded rows are all missing values — the record contains no kneel-downs, spikes or unmapped decisions on fourth down, so no row is dropped for not being a choice. Coverage does not drift: it runs from 98.33% (2015) to 99.26% (2014), a spread of $0.93\,pp$ with no trend in season ($\rho = +0.06$, $p = 0.86$), so the broad decline of the cohort thresholds across the window is not a coverage artefact. Nor is it concentrated: coverage is 98.2–99.1% across five field-position bands and 98.6–98.9% across four distance bands, with the goal line, where a punt has no defined value, at 98.7%. Of the retained rows, 19,869 fall in the innermost margin band and 2,046 in the outermost, which is why the thresholds the body uses are reweighted to equal band mass (Section 2). The clustering counts decode the record's own structure and confirm what is in it: 267 games per season through 2019, 269 in 2020, 285 from 2021, which is the regular season plus the postseason at each of the three schedule sizes.

The cohorts pass R2. A ruler used to withhold a model's threshold should survive the test it enforces. Refitting R2's own specification on the record — same five standardised covariates, errors clustered by game rather than by item — every season from 2014 to 2025 carries an independent positive `go_boost` coefficient: $+0.87$ to $+1.64$ on each season's natural distribution and $+1.30$ to $+2.21$ reweighted, $p < 10^{-18}$ in all twenty-four fits. So the cohort thresholds are not yards-to-go rules projected onto the value axis, and R5's asymmetry — a conditional gate on model thresholds and none on the ruler — is closed by measurement rather than by argument.

Two things in those fits are worth stating because they bear on the reweighting. Yards-to-go carries an independent negative coefficient in *every* cohort on the natural distribution (-0.34 to -1.30, $p < 0.01$ throughout), so the recorded policy does contain the surface cue R2 is built to detect — it simply does not consist of it. And equalising the margin bands, which the body does so that model and cohort thresholds are fitted over the same situations, also breaks much of the collinearity that carries it: after reweighting the yards-to-go term loses significance at R2's own α in four of the twelve cohorts while the value coefficient rises in all twelve. The reweighting was adopted for comparability and turns out to sharpen the fit as well.

One pattern we report and do not interpret: the value coefficient rises across the twelve cohorts ($\rho = +0.79$, $p = 0.002$ natural; $+0.77$, $p = 0.003$ reweighted) and the yards-to-go coefficient falls ($\rho = -0.80$, $p = 0.002$). Later cohorts track the value model more closely as well as more boldly. The cheapest account of that is not a change in preference at all: `nfl4th` is estimated on this same record, and a practice increasingly advised by tools built on that objective would converge on it by construction. Nothing here separates the two, so the trend is a description of the ruler and not a finding about coaching.

B WHEN THE EFFECT APPEARS

Two things that a frontier-only design cannot separate come apart across scale.

Model	Δ go|opt	Widths	Value-aligned (R2)
Qwen2.5-3B	$-4.5\,pp$	$+0.20$	no
Qwen2.5-1.5B	$-7.9\,pp$	$+0.35$	no
Qwen2.5-7B	$+4.8\,pp$	-0.21	no
Qwen2.5-14B	$-18.5\,pp$	$+0.81$	no
Qwen2.5-32B	$-38.7\,pp$	$+1.70$	no
gpt-4.1	$-58.2\,pp$	$+2.56$	**yes**
claude-sonnet-5	$-56.4\,pp$	$+2.48$	**yes**

Table 3: Persona responsiveness appears at 14B and rises monotonically. No open-weight row's optimizer or persona arm passes R2. One arm of a separate open-weight ladder run does, at a different read-out; it is reported below.

One open-weight arm passes R2, and it is reported here. The five-cohort ladder was also run on Qwen2.5-14B-Instruct in bf16 at the answer-token read-out — the model Section 5 steers, and the read-out Table 4 separates from the generated answers every row of the table above is scored on. Of that run's eleven arms ten fail the gate and one passes: the shown 2018 rung at $k = 16$, `go_boost` $\beta = +0.675$ at $p = 0.0067$ against the rule's $p < 0.01$. We report it rather than withhold it. The gate is applied per arm and without correction everywhere else in this paper, and an arm that clears a rule fixed in advance is not dropped for being inconvenient. Three things bound what it supports. It is one of eleven arms in that run and does not survive a Holm correction across them (0.0067 against $0.01/11 = 0.0009$). Its go rate is 9.8%, so the coefficient rests on about sixty going decisions in six hundred. And the other four rungs of the same ladder fail, so there is no line to fit through it and no threshold ladder to report from that model. The claim Section 5 needs is the narrower one and it holds without exception: that model's optimizer and persona arms both fail the gate, which is why the steering section measures on the go-rate axis.

Qwen2.5-32B moves 1.70 ruler widths under the persona while a multivariable logit shows its decisions carry no independent dependence on `go_boost` at all. Its univariate picture ($\rho = +0.53$, $\tau = 4.06\,pp$ $[2.72, 6.29]$) is a yardage rule projected onto the value axis, and `go_boost` correlates 0.66 with yards-to-go in this record.

Displacement does not wait for competence. A model with no measurable independent dependence on `go_boost` still shows a large displacement. One arm goes further than having no dependence and carries the wrong sign: Qwen2.5-3B under the optimizer instruction fits $\beta = -1.342$ at $p = 0.011$, so it fails R2 on sign and not only on p. We do not read that as an inverted value dependence. That arm goes on 98.8% of items, so the coefficient is fitted on the seven or so decisions that were not *go*, and a near-saturated logit is where signs become unstable rather than informative. It is reported because the gate's output is reported for every arm, and it is the reason the gate tests the sign as well as the p value.

The 14B arm was measured twice, on different hardware and by different means, and Section 5 steers the same model under a third procedure. All three report a displacement; Table 4 separates them, because an earlier reading of the third did not.

	weights	read from	decision rule	items	Δ go|opt
1	nf4	generated text	sampled, $T = 0.7$, 2 reps	300	$-18.5\,pp$
2	bf16	answer token	sampled, $T = 0.7$, 2 reps	300	$-25.7\,pp$
3	bf16	answer token	argmax, one pass	300	$-25.3\,pp$

Table 4: The same two instruction strings — `P0_optimizer` and `P2_typical_veteran` — on Qwen2.5-14B-Instruct, three ways. Rows 2 and 3 are the pair the steering sweep of Section 5 extracts its direction from, each fold measured on the half it evaluates and the two pooled, which is the whole item set. Row 3 pooled is 37 switched decisions in 146 eligible items, and it decomposes exactly: 16 of fold a's 65 ($-24.6\,pp$, the figure in Appendix J.1's gate table) and 21 of fold b's 81 ($-25.9\,pp$, the figure the steering figure's own log prints). **Those three numbers are one count read on three populations, not three measurements of one quantity**, and the optimizer arm behind them is 113 of 146, the 77.4% Section 5 quotes.

Rows 1 and 2 differ in where the letter is read *and* in precision, so they do not isolate the read-out; both sample it at temperature 0.7: the *displacement* survives that change, -18.5 against $-25.7\,pp$. Row 3 changes the decision rule rather than the read-out and survives that too, $-25.3\,pp$, with the two folds agreeing to $2.2\,pp$. **Row 3 is not a decision-rule contrast, and it is useful to say why one was once suspected here.** Read at a position carrying only 1.2% of the first-token distribution, the two prompts appear to decide identically under argmax, which invites the explanation that argmax is blind to a shift in mass that does not cross the gap between the top two options. Read at the position the prompt arms are read at, argmax and sampling agree to $0.4\,pp$, and the blindness never arises (Appendix J).

The *level* is not comparable across rows: under the optimizer instruction the same model goes 76.4% of the time generating and 77.4% reading the answer token by argmax. Displacements from rows 1 and 2 may be compared; rates may not.

C How the example blocks were built

Drawn to reproduce each cohort's go rate *within each margin band*, not merely on average. Sampling at frequency leaves about $6\,pp$ of noise on a quantity whose whole range across cohorts is $22.8\,pp$, which in a first attempt inverted two adjacent seasons; re-drawing until the order came out right would have been selecting the sample on its outcome, so the blocks used below are matched within bands rather than sampled at frequency. The examples are drawn from plays that are **not** in the 300-item evaluation set: the builder keys the evaluation items by play, removes them from the pool, prints how many it removed, and refuses to write a block file if it cannot key them all, so no model is shown an item it is about to be scored on.

The blocks were built at sixty-four examples and sixteen were delivered. `build_p5` shows the first k of a block, the run used $k = 16$, and the block file was written with $k_{\max} = 64$ seventeen minutes before the first arm ran and never rewritten, so the prefix scored here is what both models saw. The two are not the same stimulus:

	delivered, $k = 16$	built, $k = 64$
span across the five cohorts	$\mathbf{12.50\,pp}$	$21.88\,pp$
as a share of their own $21.93\,pp$	$\mathbf{0.570}$	0.998
correlation with the season rates	$+0.94$	$+1.00$
mean $\lvert\text{block} - \text{season}\rvert$	$4.16\,pp$	$0.98\,pp$
largest deviation	$9.87\,pp$ (2025)	$1.77\,pp$

The loss is not symmetric: the two boldest cohorts are shown 8.19 and $9.87\,pp$ below their seasons, and the prefix ties 2016 with 2017 and 2018 with 2022, so **five cohorts are delivered as three distinct rates**. A prefix of a band-matched sample is not itself matched, which is what `check_ladder_prefix.py` exists to measure; it was not run against this block file until after the ladder was analysed. An earlier version of this appendix reported the 0.998 as measured "from the delivered blocks". It was measured from the built ones, and the correction runs against us.

We claim no ceiling for the slope, in either direction. Not 1: the delivered stimulus does not carry the whole contrast. And not 0.570 either. The blocks are matched on the go-rate axis and the slope is fitted on the threshold axis, and dividing across the two is invalid: the two are different estimands and the map between them is not the identity. An earlier version of this appendix went on to divide anyway, observing that gpt-luna's shown slope of 0.97 against a 0.570 rate-axis contrast gives 1.70 and reading that as proof that the marginal rate cannot be the channel. It is not proof of that. A ratio above one falsifies a reader who responds in proportion to the marginal rate; it does not identify what else the block carries, because a non-linear map from displayed rate to threshold produces the same ratio with no second channel. What the delivered block might convey besides its marginal rate — the within-band composition, or the year the rung names — is not separated by any arm we ran. Refitting the threshold on the delivered blocks, which is what a ceiling in the right units would need, does not identify one: across the sixteen examples per cohort the median half-width of a block-implied threshold is $60.3\,pp$ against a cohort span of 3.65, every implied crossing falls below the lowest margin displayed, and the ceiling's own interval runs $[-17.5, +13.3]$. At sixty-four the same refit gives $[+0.43, +2.13]$ — still too wide to bound anything, and consistent with 1. The five cohorts are not a narrow slice of the record either: they span $21.93\,pp$ of the twelve cohorts' 22.77.

D RUNS

Sensitivity retention across every arm. Section 6 withdraws the claim that personas generally cost situational sensitivity. The count behind it: of the 79 arms in this paper's runs that have an optimizer arm in the same run to divide by, 30 retain 85 to 109% of its margin sensitivity — 21 of gpt-4.1's 31 against 7 of gpt-luna's 28, which is where the generality fails. Where the cost does appear it is not small: gpt-luna's shown ladder rungs retain 29 to 75%. Ratios are taken within a run, never across, for the reason Section 2 gives for pairing each arm with an optimizer condition measured beside it.

The same arm, measured five times. `P0_optimizer` on gpt-luna was run in five separate passes — the modifier ablation, the reverse anchor, the two ladders and the occupation set — each over the same 300 items at two repetitions. The five thresholds are 2.93, 3.03, 3.04, 3.11 and 3.26 *pp*: a spread of 0.33 *pp*, which is 9% of the 3.65 *pp* band the paper measures against and about a fifth of a single run's own interval. **Every one of the five intervals contains all five point estimates**, so the run-to-run variation of this estimator sits well inside the uncertainty each run already reports, and nothing in the paper turns on which pass a τ came from. It is worth saying which one did, all the same: Figure 4 draws the ablation run's 3.26, and the ladder tables quote their own runs' values, which is why the same arm reads 3.26 there and 3.04 here.

Items, clusters and rows. Three counts appear in this paper and they are one experiment, not three. The frozen evaluation set is **300 items**, drawn from the play-by-play record, and they fall into **285 source games**, which is the clustering unit: every paired bootstrap resamples games, because two items from one game are not independent. The cluster sizes are 272 games with one item, 11 with two and 2 with three, so 28 items share a game and there are 17 within-game pairs. The difference $300 - 285 = 15$ is the excess of items over games and is neither of those counts, which is why the histogram is given rather than a number derived from the two totals. **Rows** are items times repetitions, so a complete arm at two repetitions holds 600. The five-cohort run's arms hold 596 to 600: every item is present in every arm of both models, and the shortfall is repetitions the runner did not produce rather than items it lost. Rejected rows in the error log are a separate quantity again — gpt-4.1's log holds 130 for that run, of which all but 12 were produced on a retry. We did not break those 12 down by margin band, so we cannot say the loss is unrelated to the decision; 12 rows of just under 3,000 bound how much it could matter and do not characterise it. Where we do report a loss as flat across bands — the ruler's excluded rows in Appendix A — that is a statement about band totals, and a loss flat across bands can still be selective inside one.

Every figure in the body carries a run ID. The same condition measured in different runs gives slightly different values, all of them real; the gpt-4.1 optimizer arm appears below at $\tau = -0.81$, -0.89, -0.95, -0.99 and -1.01 for this reason.

The five margin bands are cut at $m = \pm 1$ and ± 4 points of `go_boost`, sixty of the 300 evaluation items in each, and the cohort thresholds the paper reports are refitted over that same distribution (Section 2).

The anchors, verbatim, and how they differ from the earlier run's. Every displacement in this paper is measured against `P0_optimizer`, *"Your sole objective is to maximize the team's probability of winning. Ignore all other considerations."* The persona, `P2_typical_veteran`, is *"Decide exactly as a typical veteran NFL head coach would."* Our earlier run froze longer forms of both. Its optimizer prefixes *"You are deciding for a professional football team"* and says *winning this game*; its persona reads *"You are an experienced NFL head coach. Decide exactly as a typical veteran NFL head coach would decide in this situation, with all the pressures and habits of the job"* — thirty words against ten. **That difference is not incidental to what this paper measures.** The earlier persona assigns the role in a sentence of its own and then asks for the convention outright, so a displacement measured under it cannot be credited to the name alone. The arms here drop both. That is why Section 2.3's 6.83 and 5.64 *pp* are not a like-for-like replication — they differ in the instruction as well as the model and the harness — and why the ablation of Section 3 starts from the bare clause, where what remains to strip is a name and its modifiers. The full set of condition strings, including the synthetic domain's, is in `src/prompts.py` in the artefact.

Why three hundred. The set is the earlier study's, so its size is inherited rather than chosen here, and reusing it rather than redrawing is what puts the two studies' numbers on the same items. What the reuse has to clear is this paper's own floor: `margin_sensitivity` returns nothing below twenty items in each extreme stratum, and sixty per band leaves about thirty in each half of the cross-fitted split, which is above that floor with room. About, because the split is one permutation of the frozen item file under a fixed seed and not a stratified draw, so the halves are not exactly even: the half Figure 3 plots holds 32 items above $+4$ and 28 below -4, and the figure reports those counts because they are what the intervals in its panel C are computed on. Both clear the floor. Two hundred items would sit exactly on it, and an uneven split would then put one band under it.

Rows lost. Two arms hold 599 and 582 of 600 rows, with the loss flat across margin strata so it does not bias the sensitivity or the threshold. Two earlier arms lost rows unevenly and were re-run rather than reported; the re-runs are what appears here.

ID	Purpose	Models	Conditions	n/arm
AB	modifier ablation	gpt-4.1, gpt-luna	P0, A1–A4	600
BO	the mirrored modifier	gpt-4.1, gpt-luna	P0, A1, A5, R2_bold	600
OC	occupation controls	gpt-4.1	P0, A1, A6–A8	600
OC2	the same, second model	gpt-luna	P0, A1, A6–A8	600
L2	two-target ladder	gpt-4.1, gpt-luna	P0, P4, P5 at $k = 4, 16, 64$	600
L5	five-cohort ladder	gpt-4.1, gpt-luna	P0, P4, P5 at $k = 16$	600
ST	steering sweep	Qwen2.5-14B-Instruct	—	150

The steering sweep's n is per fold: 150 items each, 300 when the two folds are pooled, which is the count every pooled steering number in Appendix J is computed on.

All API arms ran at 2 repetitions, `max_tokens` 512, generation mode, with the option order shuffled per item. Two models reject the `temperature` parameter and ran at the provider default; the rest were requested at 0.7. Snapshots and dates below are read from the run metadata that `runner.py` writes into every result row, not recorded by hand (`scripts/run_metadata.py`).

model	snapshot	backend	dates (UTC)
`claude-sonnet`	`claude-sonnet-5`	Anthropic API	2026-08-21 to 08-22
`gpt-4.1`	`gpt-4.1-2025-04-14`	OpenAI API	2026-08-19 to 09-16
`gpt-luna`	`gpt-5.6-luna`	OpenAI API	2026-08-22 to 08-28
`qwen2.5-1.5b-inst`	`Qwen/Qwen2.5-1.5B-Instruct`	local	2026-08-19
`qwen2.5-3b-inst`	`Qwen/Qwen2.5-3B-Instruct`	local	2026-08-19
`qwen2.5-7b-inst`	`Qwen/Qwen2.5-7B-Instruct`	local	2026-08-19
`qwen2.5-14b-inst`	`Qwen/Qwen2.5-14B-Instruct`	local	2026-08-21
`qwen2.5-32b-inst`	`Qwen/Qwen2.5-32B-Instruct`	local	2026-08-21

Table 5: Model snapshots and the dates each was run. The open-weight rows name the repository, which is the most specific identifier those runs recorded. The steering sweep of Section 5 is not listed: it writes its own per-item files rather than the runner's, and its configuration is in Appendix J.

E PREREGISTRATION: REGISTERED, EXECUTED, AND WHY THEY DIFFER

The registration fixed hypotheses, decision rules and the example construction before any ladder data existed. It specified **two** targets. Two targets determine a slope and a level exactly, so once that was clear we rebuilt the ladder with five. There was no timestamped amendment before that collection, and we do not claim one.

H6c was registered as a separation threshold on two targets and is superseded by the five-target slope; H6d fixes k and is therefore not evaluable in the body. Both are reported in their registered form in Appendix I.

Element	Registered / Executed	Reason for the difference
Targets	2014 and 2025 / 2016, 2017, 2018, 2022, 2025	Two targets leave no residual degrees of freedom; five leave three and make linearity testable
Target thresholds	season-natural / reweighted to the evaluation strata	A cohort threshold that shifts with the x distribution is not comparable with a model measured over a different one
k	4, 16, 64 / 16	Fixed to fit the five-target grid in the token budget; the k sweep survives in Appendix I
Example blocks	sampled at natural frequency / matched to each cohort's go rate within margin band	Sampling noise at $k = 64$ inverted two adjacent cohorts; re-drawing until the order came out right would have selected the sample on its outcome
H6a–H6d	unchanged	—
R2, quality floors	unchanged	—

F R2: A WORKED FAILURE

Qwen2.5-32B under the persona instruction shows $\rho = +0.53$ against `go_boost` and $\tau = 4.06\,pp$ $[2.72, 6.29]$, which looks like a policy tracking the value model. Controlling for yards-to-go, its `go_boost` coefficient falls to $+0.085$ ($p = 0.74$) while yards-to-go sits at -3.158 ($p = 8 \times 10^{-11}$): the threshold was a yardage rule projected onto the value axis. Clustering is not optional in these fits — with repetitions the rows are not independent and ordinary errors make nearly everything significant.

G THE MODIFIER ABLATION

The registered instruction carries four things at once: an instruction to adopt a role, a role that exists, and two modifiers. Stripping them one at a time, in a single grammatical frame so that sentence form is not itself a variable, gives displacement in go-rate ruler widths against the optimizer measured in the same run.

The 2×2 in full. Displacement in ruler widths, positive meaning more cautious: real coach on football $+2.48$ (claude-sonnet-5) and $+2.56$ (gpt-4.1); invented practitioner on football $+1.34$ and $+1.55$; real coach on the synthetic discipline -0.85 and $+0.00$; invented practitioner there -0.45 and $+0.09$. The synthetic residuals Section 3.1 quotes are each over the same model's football effect under the same name. **Two ratios are stable where we did not expect them to be.** claude-sonnet-5's reversal is the same share of its football effect under both names (-34.3% and -33.6%), and the invented name's own factor is nearly the same in the two domains ($1.34/2.48 = 54.0\%$ in football, $0.45/0.85 = 52.9\%$ in the synthetic one; in gpt-4.1, 60.5% in football, with the synthetic ratio undefined at a zero denominator). Neither was registered and $n = 2$; we report them because a coincidence at two decimals is worth a reader's attention either way. The synthetic optimizer arm appears in two runs and agrees to $0.6\,pp$ in both models, which is the one internal replication the 2×2 affords.

Arm	Instruction	gpt-4.1	gpt-luna
A4	a crew director	$+1.08$	$+0.53$
A1	an NFL head coach	$+1.13$	$+0.47$
A2	a *veteran* NFL head coach	$+1.41$	$+0.27$
A3	a *typical* NFL head coach	$\mathbf{+2.00}$	$\mathbf{+1.35}$
P2	a typical veteran NFL head coach	$+2.56$	—

In threshold units on gpt-4.1: the optimizer sits at $\tau = -0.81$ $[-1.41, -0.25]$ with sensitivity $+81.7\,pp$; A1 at $+2.21$ $[1.58, 2.96]$, $+81.7\,pp$; A3 at $+4.16$ $[3.35, 5.21]$, $+71.7\,pp$; A5 (*aggressive*) at -10.73 $[-14.52, -8.00]$, $+25.8\,pp$. A1 was measured in two independent runs at $\tau = 2.13$ and 2.21.

A3 and A8 are the same instruction. Both are *decide exactly as a typical NFL head coach would*, and they carry two identifiers only because they were run twice: A3 in run AB, where the question was what each modifier adds, and A8 in run OC, where the coach arm had to carry *typical* so that the three nouns differ in nothing else. A8 exists because run AB put *typical* on the baker and the nonsense token but not on the coach, which confounds the noun with a modifier worth $+1.75\,pp$ in that same run. The two runs give $\tau = +4.16$ and $+4.05$. That is an independent re-measurement of the same string, agreeing to $0.11\,pp$, against the smallest step in Table 1 of $1.59\,pp$.

The steps of Table 1, paired. Comparing two marginal intervals is a conservative test of a difference when both arms were measured on the same items. Resampling the clusters once per replicate and refitting every arm on that resample gives the difference within a draw:

Step		difference	95% CI	p
A7 − P0	*naming anybody*	$+1.12$	$[+0.65, +1.64]$	< 0.001
A6 − A7	*a real occupation*	$+2.24$	$[+1.69, +2.85]$	< 0.001
A8 − A6	*the domain's occupation*	$+1.59$	$[+1.03, +2.27]$	< 0.001
A8 − A1	*the modifier*	$+1.75$	$[+1.18, +2.45]$	< 0.001
A6 − A1	*adjective against occupation*	$+0.16$	$[-0.24, +0.58]$	0.437

Table 6: gpt-4.1, run OC, 285 clusters over the 300 items, 4,000 resamples. Pairing matters most where the marginal intervals nearly touch: the first and third steps clear the marginal comparison by 0.09 and $0.08\,pp$ and are bounded below here at $+0.65$ and $+1.03\,pp$. The last row is below the rule because it is not a step in the ladder; it is the adjective set against the occupation, and it is the one difference here that is not distinguishable from zero — $+0.16$ $[-0.24, +0.58]$, an interval $0.82\,pp$ wide, so the difference is bounded rather than merely unresolved. No equivalence margin was set in advance, so this is a difference not detected and not an equivalence established.

The second model's arms. gpt-luna, run OC2: 300 items, 2 repetitions, 100% parse, one run. Section 3 quotes the displacements; these are the thresholds they come from.

Arm	Instruction	τ	95% CI	Sens.
P0	optimizer	$+3.03$	$[2.34, 3.89]$	$72.5\,pp$
A7	a typical **zorbil**	$+4.21$	$[3.38, 5.29]$	$66.7\,pp$
A6	a typical **baker**	$+6.28$	$[5.01, 8.14]$	$49.2\,pp$
A8	a typical **NFL head coach**	$+8.18$	$[6.40, 10.64]$	$38.3\,pp$
A1	an NFL head coach	$+4.67$	$[3.72, 6.04]$	$63.3\,pp$

Table 7: The occupation controls on a second model. Every arm passes R2 (`go_boost` $\beta = +1.38$ to $+1.75$, $p < 10^{-3}$, SE clustered by item over 300 clusters), so each threshold is interpretable. The coefficients sit below gpt-4.1's $+1.76$ to $+2.78$ on the same gate, and the lowest of them is the coach arm, which is also the least sensitive.

The coach arm sits at $P = 1.57$, outside the recorded range: this model begins inside the range, so the same displacement that lands gpt-4.1 among the cohorts carries gpt-luna past all of them.

The ordering also holds on the second ruler, which needs no threshold and so does not depend on the R2 gate at all: the go rate where going is the higher-value option falls 29.3, 23.2, 16.5, 11.0% along P0, A7, A6, A8, with A1 at 21.3%. Two axes, one ordering — in this model. Appendix J.1 measures those same go rates on an open-weight model, where an unrelated occupation displaces the policy further than the domain's own — the reverse of the step this ordering rests on — which is what confines the ordering to the models it was measured on.

One further pattern in the same fits, reported as an observation rather than a test. The yards-to-go dependence this model's optimizer carries ($\beta = -1.11$, $p = 0.04$) is not significant in any of the three arms that carry *typical* ($p = 0.21$, 0.48, 0.10) and remains so in A1, which does not ($\beta = -1.03$, $p = 0.01$). That is the shape Section 4.3 reports from run L2 on other models, and it appears here without being looked for — but five arms with no correction for multiplicity is a pattern consistent with that finding, not a second demonstration of it.

The four paired differences, over the 285 clusters the 300 items fall into and 4,000 resamples:

Step		difference	95% CI	p
A7 − P0	*naming anybody*	$+1.18$	$[+0.57, +1.88]$	< 0.001
A6 − A7	*a real occupation*	$+2.07$	$[+1.13, +3.46]$	< 0.001
A8 − A6	*the domain's occupation*	$+1.90$	$[+0.51, +3.78]$	0.005
A8 − A1	*the modifier*	$+3.51$	$[+2.01, +5.51]$	< 0.001
A6 − A1	*adjective against occupation*	$+1.61$	$[+0.53, +2.94]$	0.004

Table 8: gpt-luna, run OC2. Each difference is taken within a bootstrap draw, so the item variation the two arms share does not enter it. The three ladder steps are the first three rows. The last row is the same comparison Table 6 makes in gpt-4.1, so the two models are tested the same way. Ten paired tests are reported across this appendix without correction for multiplicity; seven are at $p < 0.001$ and survive any of the usual corrections, and the three that would be worth reading carefully either way are named in the text — the $p = 0.005$ step and the $p = 0.004$ last row here, and the non-significant adjective-against-occupation contrast at $p = 0.44$ in Table 6, which a correction would only make harder to reject.

The modifier's point estimate is larger in this model than in gpt-4.1 ($+3.51$ against $+1.75$ in run OC), but those intervals overlap and the runs are not paired across models, so we do not read a difference into it.

What the baker arm says in words. The rationales carry something the rates cannot. With no baker's judgment about fourth downs to reason from, the model produces the conventional football decision and attaches the name it was given — *"Most typical coaches and bakers would punt here early in the game"*. That is the observation Section 5's account of the name as a degree of reach, rather than as content, is fitted to.

The comparison that does not need them is A6 against A1, which sets the adjective against the occupation being the right one. In gpt-4.1 that difference is $+0.16\,pp$ $[-0.24, +0.58]$, $p = 0.44$: no difference is detected between *a typical baker* and *an NFL head coach*, on an interval that excludes anything larger than about half a point in either direction. We set no equivalence margin in advance, so this is a difference not detected rather than an equivalence established. In gpt-luna the baker goes further, and paired in the same way that is $+1.61\,pp$ $[+0.53, +2.94]$, $p = 0.004$ — a difference rather than an equality. **In neither model does the right occupation without the adjective beat the wrong occupation with it**, and that now rests on a paired test on both sides: no detected difference in one model and a positive difference in the other. The ordering of the three modifier-matched nouns is untouched by this, since all three carry *typical*.

R2 passes on every arm here ($\beta = +1.10$ to $+2.60$, $p < 0.01$), so the thresholds are interpretable. The one arm without a threshold is the superlative, which saturates. `A5_aggressive_role` sits near the R2 boundary at $p = 0.0086$; see the limitations.

H Is the slope recall?

A model that simply knows more about one end of the record could produce a slope with no steering in it, recalling rather than being steered, and the two look identical in the output. The argument against this is structural, and we put it first because it does not depend on a measurement.

The described rung names no year. P4 states a rate — "goes for it about 25% of the time when going is the higher-value option" — and nothing else. No season, no era, no cue that could be differentially familiar. It nonetheless produces a slope of 0.26 in gpt-4.1 and 0.51 in gpt-luna (Table 2). Whatever carries the cohort into the policy on that rung, it is not recall of a season's figure — on this argument and the probe below, which the limitations weigh rather than treat as settled.

We also probed recall directly, asking each model for each target season's go rate when going was the higher-value option, ten times per target. The result is real and inconclusive, and we report it that way.

	2016	2018	2022	2025
truth	25.4%	30.7%	39.4%	47.4%
gpt-4.1 median	29.0	28.0	48.0	—
answered	100%	100%	100%	**0%**
gpt-luna median	30.0	32.5	55.0	—
answered	10%	40%	20%	**0%**

Table 9: Recall probe, 10 repetitions per target. 2017 omitted for space. "Answered" is the share of probes returning a number rather than UNKNOWN, which the prompt explicitly permits.

gpt-4.1 answers every probe for the seasons it answers at all, within 2.7–8.6 *pp*. gpt-luna declines 80% of the probe, so a spread across its targets would be computed from one to four answers each and we do not report one.

The probe supports the structural argument in a second way, from the shape of what gpt-4.1 recalls rather than its accuracy. Its recalled rates run 29.0 to 48.0% across 2016–2022 where the record runs 25.4 to 39.4 — a *wider* spread than the truth — and it puts 2016 above 2018 where the record has them the other way round. Its policy does the opposite of both: the slope compresses the range to 0.32 of it, and the five arms sit on a line with a residual sd of 0.15 *pp*. A policy that were recall in disguise should inherit recall's width and recall's disorder. This one inherits neither.

Neither model states a figure for 2025, the boldest cohort and one both models distinguish in the ladder. That is suggestive and not decisive: silence on the most recent season is equally what a training cutoff looks like, and the probe invites UNKNOWN rather than an estimate, so refusal and ignorance are not separable here.

What the probe measures is whether a model will *state* a statistic. It does not measure whether the cohorts' relative order is available implicitly, which is what Section 4 shows happening. Our position is that the described rung is the stronger structural argument against explicit season recall, and the probe adds a consistent but weaker observation.

I THE TWO-COHORT LADDER

Before the five-cohort design, the specification test ran on two targets (2014 and 2025) with $k \in \{4, 16, 64\}$. It is reported here rather than in the body because two targets determine a slope and a level exactly — no residual degrees of freedom, no test of linearity — so the claim the body makes is drawn from the five-cohort run instead.

Three things carry over and one does not. **Carried over:** showing separates the targets far more than describing; every shown arm but one sits conservative of the target it names, the exception being gpt-4.1's $k = 16$ rung at 2014; and error is not monotone in k — gpt-luna's 2025 arm moves 4.37, then 5.18, then 3.88 *pp* away as examples are added, and both of gpt-4.1's do the same. The one arm of the four whose error does rise monotonically is gpt-luna's 2014 one, at 5.03, 5.88 and 6.11 *pp*.

The 3.26 *pp* denominator in the table is this run's own gap between the two named thresholds, and it is not the 3.11 *pp* that Section 2's natural-distribution range (5.26 to 2.15 *pp*) implies for the same two seasons. The two-target run predates the reweighting and we have not reconciled which ruler its targets were drawn on, so the separations above are reported as percentages of the run's own gap and are not comparable with the five-cohort run's 3.65 *pp*. This is one reason the body reports only the five-cohort design. **Not carried over:** the middle rung. gpt-4.1 at k=16 separates the targets by 21% against 58% and 66% on either side of it, and retains more margin sensitivity than either. We have no account of it, and with three points we would not trust one.

What the instruction does to the fit, and where the two models diverge. Table 12 is the R2 specification refitted on every arm of this run, and it is what Section 4.3 reads. The two models are opposites on it. In gpt-4.1 the value coefficient rises at every rung and the yards-to-go coefficient stops being significant in seven arms of eight; in gpt-luna the described rungs *lower* the value coefficient and roughly double the yards-to-go one, and the 2025 described arm fails the gate because of

Arm	β go_boost	APE	Δ APE vs P0	p
P0 optimizer	$+1.46$	$+13.52$	—	—
P4 2014	$+2.25$	$+21.27$	$+7.75$ $[+1.56, +15.78]$	0.016
P4 2025	$+2.29$	$+21.84$	$+8.32$ $[+2.67, +16.18]$	0.008
P5 2014 k=4	$+2.47$	$+14.09$	$+0.56$ $[-7.24, +9.05]$	0.856
P5 2014 k=16	$+2.72$	$+23.11$	$+9.59$ $[+1.55, +18.74]$	0.023
P5 2014 k=64	$+2.97$	$+18.51$	$+4.98$ $[-2.54, +14.03]$	0.171
P5 2025 k=4	$+2.83$	$+18.90$	$+5.38$ $[-1.58, +13.73]$	0.124
P5 2025 k=16	$+3.24$	$+25.07$	$+11.55$ $[+4.03, +20.66]$	0.004
P5 2025 k=64	$+3.13$	$+24.08$	$+10.56$ $[+3.30, +18.91]$	0.003

Table 10: What the examples do to the realised weight on value, gpt-4.1, run L2. APE is the average partial effect of go_boost — the mean over the arm's rows of $\beta \cdot \hat{p}(1-\hat{p})$ — in probability points per standard deviation of go_boost ($4.75\,pp$ of win probability). Unlike β it is comparable across arms whose base rates differ, which is why Section 4.3 reads it rather than the coefficient (Allison, 1999; Mood, 2010). Neither of the two halves it replaces is evidence on its own: a falling raw go-rate gap between the extreme strata is what an intercept shift produces with β unchanged, and a rising β is identified only up to scale across arms with different base rates. Differences are taken within paired cluster bootstrap draws over the 285 clusters the items fall into, and every arm is standardised on one pooled scale so the coefficients are in the same units. Neither choice costs anything: the β column agrees to three decimals with the per-arm R2 fits over all rows in Table 12, and the APE column is a function of those same fits over those same rows, so it inherits that agreement rather than resting on a further assumption. The eight differences are reported without a correction for multiplicity, as in Appendix G: Holm keeps three at $\alpha = 0.05$ and Bonferroni two. What Section 4.3 claims is not that a particular arm clears a threshold but that the effect rises across the set and falls nowhere. Tested that way on the shared draws, the mean rise across the eight arms is $+7.34\,pp$ $[+1.91, +13.91]$, $p = 0.013$. The literal form — every arm rising inside the same draw — holds in 55% of draws, because P5 2014 k=4 straddles zero and one arm on the line drags the minimum under it however large the other seven are. **Every persona arm's APE is higher than the optimizer's and none is lower**, while margin sensitivity over the same arms runs from $78.3\,pp$ down to 48.3. The two smallest rises are 2014 at $k = 4$ and $k = 64$, which are also the two arms that lose the most sensitivity (48.3 and $50.8\,pp$): where the realised weight on value barely moves is where the policy has flattened most.

Arm	τ, 2014	τ, 2025	Sep.	% of $3.26\,pp$ gap
gpt-4.1, optimizer $\tau = -1.01$				
P4 described	1.13	0.86	0.27	8%
P5 k=4	6.14	4.26	1.88	58%
P5 k=16	4.26	3.56	0.70	21%
P5 k=64	6.25	4.11	2.14	66%
gpt-luna, optimizer $\tau = 3.11$				
P4 described	5.68	—	—	—
P5 k=4	10.29	6.52	3.77	116%
P5 k=16	11.14	7.33	3.81	117%
P5 k=64	11.37	6.03	5.34	164%

Table 11: The two-target run's thresholds. The empty cells are an arm with no threshold to report: gpt-luna's described 2025 rung fails R2 (Table 12), so the separation and the percentage that would be computed from it are withheld rather than printed, because the paper's own gate forbids reading a threshold off an arm that fails R2.

it. So "the examples are read" is one model's mechanism reported as such: both models move, and only one of them moves by attending more to value. The eighth gpt-4.1 arm is P5 2014 at $k = 64$, where yards-to-go returns at $+1.09$ ($p = 0.0006$) with the sign reversed — the optimizer goes less often on long yardage and that arm goes more, which is not the optimizer's dependence surviving but a different one appearing.

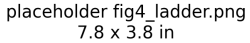

Figure 1: The two-target ladder over k, which the five-cohort run supersedes. Dashed lines and shaded bands are the two named cohorts and their intervals; error bars are per-arm cluster bootstraps. The non-monotonicity in k is visible here and is the H6d result. One arm is absent rather than unplotted: gpt-luna's described 2025 rung fails R2 in this run ($\beta = +0.75$, $p = 0.024$, the decision carried by yards to go), so its threshold is withheld here exactly as Table 11 withholds it. The figure is generated with the gate applied from the same `driver_test` output the tables use, so the two cannot drift apart.

Arm	β `go_boost`	p	β yds-to-go	p	R2
gpt-4.1					
P0 optimizer	$+1.456$	1.5×10^{-4}	-0.925	0.0008	pass
P4 2014	$+2.247$	1.5×10^{-5}	-0.586	0.091	pass
P4 2025	$+2.291$	1.3×10^{-5}	-0.609	0.070	pass
P5 2014 k=4	$+2.475$	1.7×10^{-8}	$+0.703$	0.056	pass
P5 2014 k=16	$+2.727$	2.2×10^{-5}	$+0.314$	0.315	pass
P5 2014 k=64	$+2.969$	2.5×10^{-6}	$+1.085$	0.0006	pass
P5 2025 k=4	$+2.832$	1.0×10^{-6}	-0.002	0.997	pass
P5 2025 k=16	$+3.243$	9.2×10^{-6}	$+0.247$	0.512	pass
P5 2025 k=64	$+3.136$	1.2×10^{-6}	$+0.439$	0.232	pass
gpt-luna					
P0 optimizer	$+1.784$	4.2×10^{-5}	-1.014	0.060	pass
P4 2014	$+1.125$	7.1×10^{-6}	-1.543	7.9×10^{-6}	pass
P4 2025	$+0.747$	0.024	-1.857	8.9×10^{-6}	**fail**
P5 2014 k=4	$+1.516$	1.2×10^{-4}	$+0.356$	0.234	pass
P5 2014 k=16	$+1.254$	7.9×10^{-4}	$+0.072$	0.857	pass
P5 2014 k=64	$+1.831$	2.5×10^{-7}	$+1.000$	0.0028	pass
P5 2025 k=4	$+1.864$	1.5×10^{-5}	-0.538	0.435	pass
P5 2025 k=16	$+1.830$	8.0×10^{-6}	-0.053	0.905	pass
P5 2025 k=64	$+2.085$	6.7×10^{-7}	$+0.214$	0.614	pass

Table 12: R2 refitted per arm, run L2: a logistic on standardised {`go_boost`, yards-to-go, field position, score margin, quarter} with standard errors clustered by item, 600 rows and 300 clusters per arm. The two covariates not shown are significant in most arms and are in the released `driver_test` output. R2 requires $\beta > 0$ on `go_boost` at $p < 0.01$; the one failure is the arm whose threshold Table 11 therefore withholds.

The five-cohort points. Table 2 reports a slope, an R^2 and a range of absolute errors, none of which let a reader recompute the fit or see which target carries it. The ten thresholds it was fitted to, with their intervals, are below.

Model	Named cohort τ [95% CI]	τ described (P4)	τ shown (P5)	$\lvert\text{err}\rvert$ shown [95%]
gpt-4.1, optimizer $\tau = -0.99$ $[-1.60, -0.42]$				
2016	6.10 $[5.30, 7.09]$	$+1.34$ $[+0.75, +2.05]$	$+5.35$ $[+4.33, +6.69]$	0.75 $[0.04, 2.19]$
2017	5.60 $[4.85, 6.52]$	$+1.29$ $[+0.70, +2.03]$	$+4.89$ $[+4.03, +6.03]$	0.71 $[0.04, 1.89]$
2018	4.31 $[3.86, 4.81]$	$+1.10$ $[+0.51, +1.85]$	$+4.51$ $[+3.76, +5.56]$	0.21 $[0.01, 1.39]$
2022	3.45 $[3.06, 3.88]$	$+0.57$ $[+0.04, +1.26]$	$+4.44$ $[+3.62, +5.48]$	0.99 $[0.11, 2.16]$
2025	2.45 $[2.19, 2.73]$	$+0.46$ $[-0.10, +1.18]$	$+4.04$ $[+3.17, +5.11]$	1.59 $[0.64, 2.77]$
gpt-luna, optimizer $\tau = +3.04$ $[+2.31, +3.99]$				
2016	6.10 $[5.30, 7.09]$	$+5.83$ $[+4.57, +8.09]$	$+8.07$ $[+6.44, +10.74]$	1.97 $[0.17, 4.45]$
2017	5.60 $[4.85, 6.52]$	$+5.11$ $[+4.18, +6.30]$	$+10.68$ $[+8.40, +13.73]$	5.08 $[2.68, 8.09]$
2018	4.31 $[3.86, 4.81]$	$+5.58$ $[+4.30, +7.87]$	$+8.58$ $[+6.77, +10.68]$	4.27 $[2.48, 6.50]$
2022	3.45 $[3.06, 3.88]$	$+4.27$ $[+3.17, +6.05]$	$+6.62$ $[+5.45, +8.07]$	3.17 $[1.91, 4.70]$
2025	2.45 $[2.19, 2.73]$	$+3.73$ $[+2.89, +4.94]$	$+5.87$ $[+4.87, +7.36]$	3.42 $[2.37, 4.90]$

Table 13: Every threshold behind the five-cohort fit of Table 2, with per-arm cluster bootstrap intervals; $k = 16$ throughout. Mean absolute error over the five targets is 3.43 and $0.85\,pp$ for gpt-4.1 described and shown, and 0.83 and 3.58 for gpt-luna — the two rungs swap places between the models, and the best two settings in the paper are one model's shown rung and the other's described rung, at 0.85 and 0.83. One arm to note: gpt-luna's 2017 shown rung reaches $+10.68$, the highest threshold any arm in the five-cohort run reaches, against a named target of 5.60; two arms of the two-target run go higher still ($+11.14$ and $+11.37$, Table 11). *What these two decimals do and do not reproduce.* Every slope this paper quotes for the five-cohort design — in Table 2, in the leave-one-out table and in the gate sensitivity table — is recovered exactly by refitting the five points as printed here: $+0.318$, $+0.263$, $+0.967$ and $+0.515$. The bootstrap intervals beside them are not, because each replicate refits on unrounded thresholds, and the point estimate of that fit differs in the third decimal for three of the four arms ($+0.261$, $+0.966$ and $+0.514$; the gpt-4.1 shown rung agrees at $+0.318$). The levels are quoted unrounded, and there the rounding shows one place earlier: recomputing gpt-luna's described level from these two decimals gives 4.90 against the 4.91 quoted. So a reader recomputing a slope from this table should get the paper's number, a reader recomputing a level may be $0.01\,pp$ out, and the largest of these gaps is 0.002 on a slope whose interval is 0.26 wide — no interval, verdict or ordering in the paper moves with any of them.

Five points, three residual degrees of freedom. A fit with t_3 behind every interval invites three questions, and all three are answerable from the ten thresholds above without another run.

Leave one target out. Refitting on the remaining four moves gpt-4.1's shown slope only between $+0.255$ and $+0.349$, so the compression the body reports does not rest on any single cohort. gpt-luna's moves between $+0.641$ and $+1.591$ — dropping 2016 nearly doubles what dropping 2017 halves. That is not a separate finding; it is the same finding as the standard error, arriving in a form that cannot be read as a point estimate.

Dropped	gpt-4.1 shown	gpt-luna shown	gpt-4.1 desc.	gpt-luna desc.
— (all five)	$+0.318 \pm 0.049$	$+0.967 \pm 0.450$	$+0.263 \pm 0.043$	$+0.515 \pm 0.166$
2016 ($h = 0.53$)	$+0.255 \pm 0.037$	$+1.591 \pm 0.192$	$+0.288$	$+0.498$
2017 ($h = 0.36$)	$+0.349 \pm 0.049$	$+0.641 \pm 0.352$	$+0.259$	$+0.604$
2018 ($h = 0.20$)	$+0.317 \pm 0.053$	$+0.974 \pm 0.521$	$+0.264$	$+0.522$
2022 ($h = 0.30$)	$+0.332 \pm 0.059$	$+0.902 \pm 0.573$	$+0.242$	$+0.492$
2025 ($h = 0.61$)	$+0.324 \pm 0.087$	$+0.843 \pm 0.783$	$+0.271$	$+0.415$

A test that does not use t_3 at all. Permuting which measured threshold is assigned to which named cohort gives an exact reference distribution over the 120 orderings. gpt-4.1 recovers the cohorts' rank order **exactly** on both rungs — Spearman $\rho = +1$, whose one-sided $p = 1/120 = 0.008$ is the smallest value $n = 5$ admits — and gpt-luna does not, at $\rho = +0.70$, $p = 0.12$ on the shown rung and $+0.90$, $p = 0.042$ on the described one. Permuting the slope's magnitude rather than the ranks agrees: $p = 0.008$ and 0.017 for gpt-4.1's two rungs, 0.117 and 0.075 for gpt-luna's. **The body's reading survives dropping the parametric reference distribution.**

Both sources of uncertainty at once. The five-point $\pm$ earlier versions reported is the scatter of the five measured thresholds about the line. Two sources are missing from it and they do not enter the

Figure 2: The table above, drawn. Named threshold against threshold reached, five cohorts, both rungs, both models. The dotted diagonal is exact placement; bands are the cohorts' own intervals and error bars are per-arm cluster bootstraps. **The wedge around each fit is the five-point least squares standard error** pivoted about the mean target, which is the quantity Table 2 no longer reports: against the paired bootstrap it understates gpt-4.1's line and overstates gpt-luna's, so read the wedges as the scatter of five points about a line and the intervals in Table 2 as the uncertainty of the fit. The figure is in the appendix because every number in it is tabulated here and in Table 2, and because a wedge the paper has superseded does not belong in the main text.

same way. The named thresholds on the other axis carry their own intervals, measured on different data; and the five measured thresholds are themselves estimates, but from the *same* 300 items, so their marginal intervals share the item variation and that component cancels in a slope. Propagating the model side as independent noise would double-count it — the residuals about the line are an order of magnitude smaller ($0.13\,pp$ rms) than the marginal intervals ($\pm 1.2\,pp$), which is what the cancellation looks like from outside — and adding the two sides in quadrature afterwards assumes what the cancellation denies.

The estimator that avoids both errors resamples the clustering unit once per replicate and refits every arm inside that one draw, the discipline Appendix G applies to differences between arms, here applied to a line through five of them. The 285 source games are resampled, all five arms refitted on that draw with `fit_threshold`, and each cohort's named threshold redrawn from its own bootstrap interval in the same replicate; 4,000 replicates, the point estimates unchanged from the least squares fit by construction. Table 2 reports the result:

	point	paired bootstrap 95%	five-point ±, as t_3 95%
gpt-4.1 shown	$+0.318$	$[+0.094, +0.577]$	$[+0.162, +0.474]$
gpt-4.1 described	$+0.263$	$[+0.141, +0.404]$	$[+0.126, +0.400]$
gpt-luna shown	$+0.967$	$[+0.500, +1.559]$	$[-0.465, +2.399]$
gpt-luna described	$+0.515$	$[+0.252, +0.828]$	$[-0.013, +1.043]$

The correction does not run in one direction. It widens gpt-4.1's shown interval by half again and narrows gpt-luna's to a third, because gpt-luna's five-point scatter is dominated by one arm — its 2017 target, whose landing error of $5.08\,pp$ is the largest in the run — while gpt-4.1's five points sit close enough to the line that the arms' own measurement error is what matters. The five-point $\pm$ was not a conservative approximation; it was a different quantity. Under the bootstrap gpt-4.1's shown slope still excludes 1 ($P = 0.0003$ of replicates at or above it) and all four arms exclude zero (no replicate of 4,000 in any arm), which is the one comparison in the body that changes: Section 4 previously reported gpt-luna's ordering as not measurable. The levels move the same way — gpt-4.1's shown level, 4.65, separates from the correct 4.38 on the five-point fit and not under the bootstrap $[3.84, 5.69]$.

How the draws are taken, in full. Per replicate: the 285 source games are resampled with replacement to 285; every arm is refitted on that one draw with `instrument.fit_threshold`; and each cohort's named threshold is redrawn once, shared by the arms in that replicate, as a normal draw centred on its point estimate with standard deviation $(\text{ci_hi} - \text{ci_lo})/(2 \times 1.96)$. That last step is an approximation and we state it rather than hide it: a bootstrap interval is not symmetric in general, and matching a normal draw to its half-width discards whatever asymmetry the cohort's own resampling produced. No stratification is applied, the clustering unit is the same for every arm, and a replicate is discarded if any arm's resample admits no finite threshold — across the six runs reported here, **no replicate was discarded**, so every interval rests on all 4,000. The draws themselves are written out per replicate, so any further contrast can be taken inside the draw rather than between two intervals.

Six quantities, kept apart. Earlier versions of this paper used "reproduces the ordering" for two of the rows below, which are not the same claim and do not have the same answer. Intervals are the paired bootstrap above; ρ and its p are the exact permutation test; the level is compared with the correct $4.38\,pp$.

	gpt-4.1		gpt-luna	
	shown	described	shown	described
slope on the named threshold	$+0.318$	$+0.263$	$+0.967$	$+0.515$
95%	[.09, .58]	[.14, .40]	[.50, 1.56]	[.25, .83]
excludes 0 / excludes 1	yes / yes	yes / yes	yes / **no**	yes / yes
rank order recovered (ρ, p)	$+1$, .008	$+1$, .008	$+0.70$, .12	$+0.90$, .042
level at the mean target	4.65	0.95	7.96	4.91
separates from 4.38?	**no**	yes	yes	**no**
mean |landing error|	0.85	3.43	3.58	0.83
95%	[.56, 1.44]	[2.74, 4.08]	[2.24, 5.27]	[.47, 2.12]
targets inside their own interval	3/5	—	0/5	2/5
targets *shown to miss*	2/5	5/5	5/5	1/5
under Holm at 0.05	1/5	5/5	5/5	1/5

The dashes are quantities we did not compute, not zeros. Reading down a column: a model can track the cohorts with a positive slope, order them correctly, sit at the right average level, and still place none of the five — and the two models differ in which of those they achieve.

What the landing errors estimate. For cohort j the landing error is the signed difference $\delta_j = \tau_{\text{arm}}(j) - \tau_{\text{cohort}}(j)$ between the threshold the arm reaches and the threshold that cohort's own plays imply. Three different quantities in the tables above are built from it, they answer different questions. All three are defined with the counterfactual value model held fixed: nothing here resamples it, so no interval in this appendix carries its error. A difference between two arms measured against the same fixed value model is insensitive to a common *shift* in it and a level is not; neither is insensitive to a common *scale* error, which multiplies a difference as it multiplies a level. The bootstrap supplies the rest of the variation on both sides at once — the arm side by resampling the 285 source games, the cohort side by the normal draw above, taken independently across the five cohorts and shared across arms inside a replicate.

One. The mean level error, which the table already reports once. The level at the mean target is the least squares fit evaluated at $\bar{x} = 4.38\,pp$, and for a fit with an intercept the mean fitted value equals the mean measured threshold, so **the level minus** 4.38 **is exactly** $\bar{\delta}$, not an approximation to it: -3.43, $+0.27$, $+3.58$ and $+0.53\,pp$ for gpt-4.1 described and shown and gpt-luna shown and described. The row *separates from* 4.38*?* is therefore the test of whether the mean level error is zero, and gpt-4.1's shown interval $[3.84, 5.69]$ is the interval $[-0.54, +1.31]$ on that error. Two of the four arms have a mean level error not separated from zero. That is a statement about the average of five signed errors and not about any single target: an arm whose errors cancel has a small average and may still miss every cohort. What the individual targets do is the third quantity below.

Two. The mean absolute error, which is a different object and is biased away from zero. $\mathbb{E}|\delta| \geq |\mathbb{E}\delta|$ for any δ, with equality only when the sign does not change, so an arm that placed every target exactly would still report a positive mean absolute error, of a size set by its own measurement noise

rather than by any mismatch. Its interval, taken over the same draws, is $[0.56, 1.44]$ and $[2.74, 4.08]$ for gpt-4.1 shown and described and $[2.24, 5.27]$ and $[0.47, 2.12]$ for gpt-luna shown and described. **The two smallest are not separated from each other by anything**: 0.85 $[0.56, 1.44]$ against 0.83 $[0.47, 2.12]$, on different models and opposite rungs. Neither the column nor the pair of best settings is a ranking.

The per-target intervals say where the noise floor actually is. Table 13 now carries them for the shown rungs. For gpt-4.1's shown rung the intervals on $|\delta_j|$ reach down to 0.04, 0.04, 0.01 and $0.11\,pp$ at 2016, 2017, 2018 and 2022: at four of the five targets the arm is not shown to have missed at all. A percentile interval on $|\delta_j|$ is bounded below by zero and so cannot exclude it, which is why the test that matters is how close the lower bound comes: four of the five reach within $0.11\,pp$ of the floor. Only 2025, at 1.59 $[0.64, 2.77]$, does not — and it is the target the signed errors below identify as the overshoot. The described rungs behave in the two ways the shapes predict: every one of gpt-4.1's five is clear of zero (1.30 at the narrowest), and none of gpt-luna's five is, its lower bounds running 0.03 to $0.37\,pp$.

Three. The per-target test, on the signed error. An interval on $|\delta_j|$ cannot exclude zero, so the question of whether an arm missed a target has to be asked of δ_j itself. The same draws answer it: each replicate already holds $\delta_j = y_j - x_j$ before the absolute value is taken, so the interval below is the same bootstrap, the same refit and the same cohort draw as every other number in this appendix.

signed δ_j, 95%	2016	2017	2018	2022	2025
gpt-4.1 shown	-0.75	-0.71	$+0.21$	$\mathbf{+0.99}$	$\mathbf{+1.59}$
	$[-2.17, .79]$	$[-1.89, .66]$	$[-.73, 1.39]$	$[+.02, 2.16]$	$[+.64, 2.77]$
gpt-luna described	-0.27	-0.49	$+1.28$	$+0.83$	$\mathbf{+1.28}$
	$[-1.86, 2.07]$	$[-1.74, .91]$	$[-.16, 3.48]$	$[-.39, 2.63]$	$[+.36, 2.49]$

Bold marks an interval clear of zero. The two one-signed arms are not tabulated because every one of their five is: gpt-4.1's described rung misses all five low, from -4.77 to -1.99, and gpt-luna's shown rung misses all five high, from $+1.97$ to $+5.08$, each at $p < 0.001$ except gpt-luna's 2016 at $p = 0.045$. **Every one of the four settings is shown to miss at least one target, and that survives a Holm correction over each arm's five**, which is the form in which the body's claim about placement should be read. The two arms that miss least miss at the same end and in the same direction — 2025, the most aggressive cohort, overshot by $+1.59$ and $+1.28$ — which is what a slope below one does to the bottom of the range.

What this is not. A target whose interval covers zero is *not shown to miss*; it is not thereby placed. Establishing placement needs an equivalence margin and this paper sets none, so no target here is established to sit at its cohort's threshold. The count of targets falling inside the *cohort's* interval, which the table above also reports, is a third thing again: it ignores the arm's own uncertainty, which is the wider of the two. It agrees with the signed test on three of the four arms and not on gpt-luna's described rung, where it counts $2/5$ inside while the signed test shows $1/5$ missed — the remaining two are outside the cohort's interval and still not shown to miss.

What the first two disagree about. For a one-signed arm the mean absolute error equals the magnitude of the mean level error, and both of the large arms are one-signed: gpt-4.1 described undershoots all five and gpt-luna shown overshoots all five, at 3.43 and $3.58\,pp$ either way. The two arms with the smallest mean absolute error are exactly the two whose signed errors change sign. gpt-4.1's shown rung runs -0.75, -0.71, $+0.21$, $+0.99$, $+1.59$ across 2016 to 2025, crossing zero between 2017 and 2018 — which is what a slope below 1 looks like target by target, undershooting the conservative cohorts and overshooting the aggressive ones — and gpt-luna's described rung runs -0.27, -0.49, $+1.27$, $+0.82$, $+1.28$. For those two the mean level error ($+0.27$ and $+0.53$) is well inside the mean absolute error (0.85 and 0.83), and the gap is sign cancellation, not spread: $\overline{|\delta|} - |\overline{\delta}|$ is twice the smaller of the two signed sides' means, $0.58\,pp$ here, and it would be zero for any arm whose errors share a sign however widely they scattered. **The level row alone would credit these two arms with an average placement no single target of theirs achieves, and the absolute row alone would hide that the other two arms' errors are an offset rather than a spread.** Both are reported for that reason, and we claim the ordering of neither as a ranking of how well placed an arm is.

The two rungs, differenced inside one draw. Table 2's intervals are per rung, and differencing two intervals is not a test of a difference. Refitting all ten arms inside one game draw, with the cohort side drawn once and shared, gives P5 − P4 directly:

P5 − P4, paired	gpt-4.1	gpt-luna
slope	$+0.057$ $[-0.202, +0.324]$	$+0.452$ $[+0.001, +1.047]$
level at the mean target	$+3.694$ $[+2.959, +4.621]$	$+3.058$ $[+2.050, +4.266]$
mean |landing error|	-2.579 $[-3.214, -1.534]$	$+2.754$ $[+1.305, +3.958]$

The slope difference between the rungs is not established in gpt-4.1 and is marginal in gpt-luna, whose interval's lower bound is $+0.001$; we report the bound rather than the verdict it technically licenses. **The level difference is decisive in both** — the shown rung sits three to four points more conservative than the described one — **and the landing error separates in opposite directions**: showing the cohort's decisions moves gpt-4.1 $2.58\,pp$ closer to its targets and gpt-luna $2.75\,pp$ further away, with neither interval touching zero. That is the pairing-off Section 4 describes, measured as a difference rather than read off two columns.

The arms in the two-target run used example blocks drawn at natural frequency rather than policy-matched, so their landing thresholds differ from the five-cohort run by 1.3 to $1.5\,pp$ for the same cohort. The composition of the block moves the landing point by about as much as a cohort's worth of difference — which is informative, and is why the body reports the five-cohort design, whose blocks were built matched within margin bands. What that design *delivered* is the first sixteen of each block, and a prefix of a matched sample is not itself matched; Appendix C measures how far it fell short.

What the gate's threshold is worth. $p < 0.01$ is a rule this programme wrote, applied to the programme's own arms, so it should be perturbed rather than trusted. Refitting the five-point lines under a gate ten times tighter costs gpt-4.1 nothing on either rung and gpt-luna nothing on its shown rung: the *largest* p behind any arm in those three fits is 7.0×10^{-5}. **The result the body reports — one model tracking at about a third of the cohorts' scale, the other at a scale indistinguishable from correct — is invariant to the gate.**

Gate	gpt-4.1 desc.	gpt-4.1 shown	gpt-luna desc.	gpt-luna shown
$\alpha = 0.01$ (the gate used)	5 pts, $+0.263$ level $+0.95$	5 pts, $+0.318$ level $+4.65$	5 pts, $+0.515$ **level** $+4.91$	5 pts, $+0.967$ level $+7.96$
$\alpha = 0.005$	unchanged	unchanged	**4 pts**, $+0.492$ **level** $+5.06$	unchanged
$\alpha = 0.001$	unchanged	unchanged	**2 pts, not fittable**	unchanged

One fit is gate-sensitive, and it is one of the two arms whose mean level is not distinguishable from the correct 4.38 — gpt-luna's described rung, at 4.91 $[3.95, 6.37]$, the other being gpt-4.1's shown rung at 4.65 $[3.84, 5.69]$ (Table 2; the five-point ± 0.07 that separated the second from 4.38 was the line's scatter alone). At $\alpha = 0.005$ its 2022 rung fails, four points remain, and the level moves to $+5.06$, which would not pass. Section 4 already declines to read that arm as a placement — it calls it a crossing point — and this is the measurement behind that sentence. We report it rather than re-running the arm: one arm sits at $p = 0.0098$ against a threshold of 0.01, and re-measuring an arm because it is near a boundary, then keeping whichever answer comes back, is selection on the outcome.

Every arm the body reports passes R2. Table 14 refits the gate on all twenty-two arms of the five-cohort run, because a threshold in Table 2's fit would not be reportable if its arm failed, and one arm of the superseded two-target run does fail (Table 12). None here does. Four things in it are stated plainly.

The gpt-4.1 pattern of Section 4.3 holds, and in the quantity that section says to read rather than in the coefficient: the average partial effect rises from $+15.33\,pp$ in the optimizer to $+17.01$–$+23.05$ across all ten persona arms, every one of them above it (Table 15). The coefficient rises as well — $+1.67$ to $+2.08$–$+2.41$ described and $+2.11$–$+3.02$ shown — but that is the comparison across

base rates Section 4.3 declines to make, so it is reported and not relied on. The optimizer's negative yards-to-go dependence (-0.86, $p = 0.0014$) is not significant at R2's own α in any persona arm but one, where it returns at $+0.91$ with the sign reversed. That is the second such arm across the two runs, and both are reversals rather than survivals.

The gpt-luna pattern holds too, and it is the opposite one in both quantities: `go_boost` falls from the optimizer's $+1.49$ in four of its five described rungs, to between $+0.95$ and $+1.17$, and rises in the fifth ($+1.57$, the 2017 target), while yards-to-go strengthens from -1.24 to as much as -1.64, and every one of its ten arms sits *below* the optimizer's average partial effect of $+15.83\,pp$, from $+5.50$ to $+15.54$ (Table 15). Naming a cohort in words moves this model further onto the surface cue, and showing it the cohort's decisions does not.

The closest any body arm sits to the R2 boundary is gpt-luna's described 2022 rung, at $p = 0.0098$ against a threshold of 0.01. It passes, and it is one of the five arms behind one of the two non-separating mean levels in Table 2. We would not want that read as a strong result resting on a strong fit.

The gate's clustering unit does not move it. R2 clusters by item, and the paired bootstrap behind Table 2 resamples source games, so the obvious question is whether the gate survives the unit the bootstrap treats as the independent one. Refitting all 22 five-cohort arms with the standard errors clustered by game (285 clusters rather than 300 items) leaves every one of them passing, moves five arms' p up and seventeen down, and the largest change in either direction is a factor of 1.03. The boundary arm goes from 0.00980 to 0.00979. The reason the two agree is arithmetic rather than fortunate: only 15 of the 300 items share a source game with another (Appendix D), so the correlation the game unit adds involves at most 15 pairs. We keep the item gate, which is the registered one; changing the unit after seeing which arms each version removes would be choosing the rule by its outcome.

And the same condition drifts across runs, as Appendix D says it does: gpt-4.1's optimizer carries $+1.67$ here against $+1.46$ in the two-target run. Section 4.3 quotes the two-target numbers and labels them, so the two are not to be differenced.

J THE STEERING SWEEP IN FULL

A first pass at this sweep was measured at the wrong position and every number it produced was discarded. What follows is the corrected run; the defect is stated first, because it is a hazard of the method rather than an accident of this run, and the guard it motivated is now part of the procedure.

The read-out defect. `steer.py` built its own tokenizer rather than going through the backend the rest of the paper uses, and so read the answer letters two ways the rest of the paper does not: it tokenised the bare prompt where every other measurement applies the model's chat template, and it took the leading-space `" A"/" B"/" C"` ids where the capability floor takes the bare ones. Measured as a 2×2 on 24 items, the templated cell with bare ids carries 100.0% of the first-token probability mass and puts the two source prompts $26.0\,pp$ apart; the sweep's cell carries 1.2% and puts them $4.5\,pp$ apart when sampled and 0.0 under argmax. Reading three log-probabilities is a decision measurement only when those tokens hold the mass; below that it compares three renormalised tails, and a comparison of tails can come out anti-correlated with anything. **Every number in the previous version of this appendix, the sweep row of Table 4, and the layer selection itself were produced at that position and are withdrawn.** The cached persona direction was extracted there too and is not reused: every run below re-extracts it. The four control vectors are a separate case and the released logs say so — they print *reusing* a cached `.npy` — because a control has to be one fixed vector across its own settings or it is not a control. They are built once by `make_control_direction.py` from the re-extracted direction, not from the withdrawn one. The sweep now refuses to start below 50% mass, the guard the capability floor has always had, which costs 300 forward passes rather than two GPU hours.

The design. Two folds with a fixed seed: fold a extracts the direction on the first half and evaluates the second, fold b the reverse. The halves are disjoint and exhaust the item set, so pooling their per-item decisions is the 300-item measurement rather than an average of two. Pooling is only legitimate if the two folds applied nearly the same vector, which is checked and not assumed: the two directions

Arm	β go_boost	p	β yds-to-go	p
gpt-4.1				
P0 optimizer	$+1.668$	1.6×10^{-4}	-0.858	0.0014
P4 2016	$+2.405$	5.5×10^{-6}	-0.267	0.413
P4 2017	$+2.393$	5.3×10^{-6}	-0.414	0.216
P4 2018	$+2.194$	7.5×10^{-6}	-0.486	0.153
P4 2022	$+2.197$	8.2×10^{-6}	-0.690	0.038
P4 2025	$+2.082$	7.9×10^{-6}	-0.694	0.029
P5 2016	$+2.771$	9.6×10^{-9}	$+0.912$	0.0033
P5 2017	$+2.989$	4.8×10^{-6}	$+0.640$	0.078
P5 2018	$+3.016$	5.2×10^{-7}	$+0.723$	0.085
P5 2022	$+2.968$	2.4×10^{-9}	$+0.654$	0.074
P5 2025	$+2.107$	3.4×10^{-5}	-0.408	0.482
gpt-luna				
P0 optimizer	$+1.488$	3.9×10^{-4}	-1.243	0.0057
P4 2016	$+1.166$	0.0036	-1.337	0.0034
P4 2017	$+1.572$	1.4×10^{-8}	-0.534	0.073
P4 2018	$+0.963$	0.0036	-1.609	7.1×10^{-4}
P4 2022	$+0.952$	$\mathbf{0.0098}$	-1.638	2.6×10^{-4}
P4 2025	$+1.050$	9.2×10^{-4}	-1.543	2.4×10^{-5}
P5 2016	$+2.407$	1.6×10^{-5}	$+0.779$	0.058
P5 2017	$+1.001$	7.0×10^{-5}	-0.555	0.075
P5 2018	$+1.726$	1.5×10^{-6}	-0.149	0.723
P5 2022	$+2.307$	8.4×10^{-7}	-0.301	0.732
P5 2025	$+2.031$	3.7×10^{-5}	-0.421	0.565

Table 14: R2 refitted per arm, the five-cohort run, $k = 16$ throughout: a logistic on standardised {go_boost, yards-to-go, field position, score margin, quarter} with standard errors clustered by item, 596–600 rows and 300 clusters per arm. **All twenty-two arms pass** ($\beta > 0$ at $p < 0.01$), so every threshold in Table 2 and Table 13 is reportable. The bold p is the narrowest margin in the body. gpt-luna's weakest shown rung on this coefficient is 2017 ($+1.00$), which Section 4 independently identifies as its least sensitive and most displaced arm.

agree at cosine $+0.9975$ (norms 19.55 and 19.62, activation norms 127.7 and 127.8). Where a pooled figure is quoted below the per-fold pair is given beside it, because two folds disagreeing is a result about the direction's stability and not noise to average away.

The coefficient scale. rel is the added vector's norm as a fraction of the measured activation norm at the steered layer, so the same rel is the same relative perturbation at any layer and for any direction — which is what makes the control arms below comparable to the real one without rescaling. At layer 30, rel $= -0.90$ is an absolute coefficient of -5.88, computed from the unrounded norms.

The dial, under both read-outs. Pooled, the go rate falls monotonically from 89.7% at rel $= -0.90$ to 26.0% at $+0.60$ — eight of the ten non-degenerate settings, $63.7\,pp$ — and turns back outside that run, at 72.6% and 46.6%. Two settings are dropped as saturated above 90%. Re-read with the same answer token *sampled* at $T = 0.7$, two draws per item, the same twelve settings give $62.3\,pp$ over seven of nine, with three saturated. The two read-outs agree to 2% of the span. **The read-out is not what sets the width of the intervention**, which was established rather than assumed: argmax discards any shift in mass that does not cross the gap between the top two options, and could in principle have shrunk the span or enlarged it.

Cohort coverage. Four of the twelve cohorts are matched within $2\,pp$ under either read-out. The run's floor is 26.0% (argmax) and 25.7% (sampled), so 2014–2016 lie below it entirely and their nearest setting is the end of the run rather than a landing; 2023–2025 sit 7 to $10\,pp$ from theirs. One decision out of the 146 pooled items where going is the better option is $0.7\,pp$, so sub-$2\,pp$ landing errors are not distinguishable from exact and should not be read as precision. On a grid whose settings bracket a target, how often a landing comes within a given tolerance is a property of the

Arm	APE	Δ APE vs P0	95% CI	p
gpt-4.1: every arm above the optimizer				
P0 optimizer	$+15.33$	—	—	—
P4 2016	$+22.77$	$+7.44$	$[+1.29, +15.67]$	0.013
P4 2017	$+21.81$	$+6.47$	$[+0.47, +14.86]$	0.033
P4 2018	$+21.20$	$+5.87$	$[+0.06, +12.83]$	0.049
P4 2022	$+20.32$	$+4.98$	$[-0.19, +11.32]$	0.061
P4 2025	$+19.84$	$+4.50$	$[-0.54, +10.47]$	0.068
P5 2016	$+20.69$	$+5.36$	$[-2.35, +13.35]$	0.164
P5 2017	$+21.87$	$+6.54$	$[-1.65, +15.29]$	0.116
P5 2018	$+23.05$	$+7.71$	$[+0.29, +16.06]$	0.040
P5 2022	$+22.58$	$+7.24$	$[-0.51, +15.53]$	0.061
P5 2025	$+17.01$	$+1.68$	$[-6.54, +9.09]$	0.668
mean across the ten		$\mathbf{+5.78}$	$[+0.56, +11.69]$	$\mathbf{0.032}$
gpt-luna: every arm below it				
P0 optimizer	$+15.83$	—	—	—
P4 2016	$+10.75$	-5.08	$[-13.01, +1.79]$	0.149
P4 2017	$+15.54$	-0.29	$[-7.00, +5.97]$	0.897
P4 2018	$+9.32$	-6.51	$[-13.26, -0.40]$	0.035
P4 2022	$+10.53$	-5.30	$[-10.81, +0.22]$	0.059
P4 2025	$+11.19$	-4.64	$[-11.91, +0.75]$	0.105
P5 2016	$+12.54$	-3.29	$[-11.27, +2.36]$	0.240
P5 2017	$+5.50$	-10.33	$[-18.90, -3.70]$	< 0.001
P5 2018	$+9.51$	-6.32	$[-15.47, +0.82]$	0.091
P5 2022	$+12.62$	-3.21	$[-12.22, +3.64]$	0.407
P5 2025	$+13.20$	-2.63	$[-10.34, +2.16]$	0.284
mean across the ten		-4.76	$[-11.13, +0.09]$	0.056

Table 15: Average partial effect of `go_boost` on the five-cohort run, both models, in probability points per pooled standard deviation of `go_boost` ($4.75\,pp$). Same estimator as Table 10: one pooled standardisation, differences taken inside paired cluster bootstrap draws over the 285 clusters the 300 items fall into. The two counts are the same data: every arm holds all 300 items, and 15 of them share a source game with another, so the resampling unit is 285 while the item count is 300 per arm and the row count is 596–600 (Appendix D). **The two models reverse each other in the quantity Section 4.3 reads**, not only in the coefficient: all ten of gpt-4.1's arms sit above its optimizer and all ten of gpt-luna's sit below. The strength is not symmetric. gpt-4.1's mean rise clears zero; gpt-luna's mean fall does not ($p = 0.056$), so its reversal is established in the ten point estimates and not in the mean. Nor does any single gpt-4.1 arm survive a Holm correction over the ten tests, which is why the mean and not the count is what Section 4.3 quotes. One thing the column settles in passing: the shown rungs are not ordered above the described ones here either — the largest rise ($+7.71$) and the smallest ($+1.68$) are both shown arms — which is H6b failing on a third quantity.

grid's spacing as much as of the policy — a coarse grid can bracket a target and miss it on both sides — so the count of landings here is not a measurement of steerability; the columns that are not are sensitivity and agreement at those settings, and none of the four matched settings clears the $15\,pp$ sensitivity floor.

Sensitivity, and why it cannot carry a positive claim. Sensitivity is the go rate above $+4\,pp$ of margin minus the rate below $-4\,pp$. Pooling both folds puts sixty items in each tail rather than thirty, which narrows each interval by about $\sqrt{2}$ and not by half. Every setting is negative — the model goes more often when going is worse — and **no setting's interval lies above the $15\,pp$ floor, under either read-out**. It also falls further as the coefficient rises: $\rho = -0.80$ ($p = 0.002$) under argmax and -0.90 ($p < 0.001$) sampled. The individual paired differences against `rel` $= 0$ all contain zero, so what is established is the ordering and not any single step.

There is no value dependence to move. R2 fails at all twelve settings under both read-outs, and so does the unsteered policy: pooled, its `go_boost` coefficient is -0.019 ($p = 0.92$) under argmax and -0.027 ($p = 0.88$) sampled. **The two halves of the item set return $+0.278$ and -0.403**

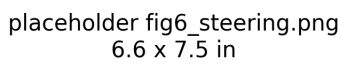

Figure 3: The corrected steering sweep on the half the steered layer was not selected on. Each panel carries one claim from Section 5 that a table states only by asking the reader to hold two columns against each other. The run shaded in A spans $67.9\,pp$ over seven of the nine non-degenerate settings; Section 5 quotes the two folds pooled, $63.7\,pp$. The faint line is the same direction on the selection half, where the same coefficients produce rates up to $28\,pp$ different for directions agreeing at cosine $+0.9975$: what differs between them is the item sample, not the intervention.

for that same unsteered policy, neither significant, which is why the coefficient is quoted pooled and why no half-sample p-value for the *unsteered* fit is relied on anywhere: a quantity whose sign reverses between the two halves is not measured on either. Appendix J.1's gate table prints them — $p = 0.33$ on fold a — because that table reports every arm of that run as it stands, and neither figure supports a claim about this policy's value dependence. The persona prompt fails R2 as well, on one half at $\beta = -0.018$, $p = 0.93$. The go rate where going is clearly worse and the rate where it is better move together at $\rho = +0.98$, which is what scaling a response bias produces and is the single most compact statement of what the coefficient does.

The layer was selected twice, and the first re-selection was wrong. The original choice scored seven layers — 12, 24, 30, 33, 36, 39, 42 of 48 — in two passes, the second chosen after seeing the first, so it was adaptive and is disclosed as such. That selection was made at the defective read-out and had to be redone. The first redone selection ranked layers by the width of their excursion and chose layer 24, **which was an error in the criterion, not in the arithmetic**: width is blind to sign, and the persona prompt lowers this model's go rate, so a layer whose rate *rises* with the coefficient scores well while moving the policy away from the persona the direction was extracted from. Layer 24 is such a layer, on both folds. Scoring instead by the width of the excursion *in the direction the prompt moves*, on the fold that does not evaluate the reported half, leaves two eligible layers of the seven — 30 at $53.8\,pp$ and 33 at $23.1\,pp$ — and disqualifies the other five, at which adding the persona-minus-optimizer difference moves the policy away from the persona. **That five of seven layers invert the behavioural sign of a difference-of-means direction is a result in its own right, and a fact about contrastive activation addition rather than about this model.** Layer 24's sweep is retained as an observation: it is the only intervention anywhere in this paper that raises margin sensitivity by an established amount ($+25.0\,pp$ against unsteered, $[+8.3, +41.7]$, at $\texttt{rel} = -0.90$), and it does so while moving away from the persona and collapsing the go rate to 13.7%, at an absolute level of $+10.0\,pp$ that still does not reach the floor.

Agreement with the persona prompt, and the control that explains most of it. Item-level agreement is argmax-based and taken one decision per item, so it measures the intervention and not the sampler. It climbs monotonically over ten consecutive settings, from $\kappa = +0.426$ unsteered — which is what the two prompts already share — to $+0.605$ at $\texttt{rel} = +0.20$. Paired over a cluster bootstrap resampling games, that is $+0.179$ $[+0.097, +0.260]$, and it survives a Holm correction over the eleven non-zero settings ($p = 0.0027$); three settings clear 0.05 adjusted and seven are established *worse* than not steering, which is the negative-coefficient half of the same curve.

A raw gain of that shape has an obvious alternative explanation: the persona prompt's go rate is lower, so a perturbation that moves the rate toward the persona's can agree with it more for that reason alone, and κ's chance term corrects for matching the rate on average rather than on the same items. The pattern is not a law: the same controls swept wide enough *lose* agreement, and two settings at the same rate differ in κ, both of which this appendix reports below. Four norm-matched control directions test it — two permutations of the real vector's coordinates, one sign flip of its magnitudes, one isotropic draw, all at the identical norm and all within $|\cos| < 0.04$ of the real direction — swept at the same layer and pooled over the same items:

direction	go rate at $+0.40$	rate moved	$\Delta\kappa$
isotropic	76.7%	$-0.7\,pp$	$+0.029$
shuffle, seed 1	71.9%	$-5.5\,pp$	$+0.036$
sign flip	66.4%	$-11.0\,pp$	$+0.117$
shuffle, seed 2	65.1%	$-12.3\,pp$	$+0.109$
persona difference	32.9%	$-44.5\,pp$	$+0.126$

Across those twelve control settings $\Delta\kappa$ is a straight line in the rate each one moved ($r = +0.963$, about $+0.008$ of κ per point). **The confound is real and it is large.** That line is the rising limb of something with a top, which a second pass established by sweeping four controls out to $\texttt{rel} = +1.60$, far enough to carry the rate past the persona's own 52.1% (76 of the 146 items with a positive value gap, against the optimizer arm's 113). Over the wide grid the correlation with rate moved falls to $r = +0.62$ and the governing quantity is instead the distance between the control's rate and the persona's ($r = -0.61$): $\Delta\kappa$ peaks near it and collapses beyond, to $+0.038$ at 40.4% and -0.111 at 26.7% — worse than not steering. The ranking among controls does not survive the extension either. The isotropic draw is the weakest control on the narrow grid ($+0.029$) and the strongest on the wide one ($+0.155$ over unsteered, at $\texttt{rel} = +0.80$). Further out it keeps a policy while the shuffles lose one: at $\texttt{rel} = +1.20$ the isotropic control still agrees with the persona prompt at $\kappa = +0.317$ where a shuffle has fallen to $+0.009$. **Those two are different quantities** — the first a difference against not steering, the second a raw agreement — and the isotropic draw's own difference has by then gone negative (-0.109). So there is no one direction to call the null.

What survives the confound is measured by pairing each real setting with a control setting at the same rate and taking the double difference within one bootstrap draw. On the narrow grid the

nearest available pair was at 65.1%, one setting below the real direction's κ peak, and there the real direction's excess over the shuffle is $+0.058$ $[+0.009, +0.109]$ and over the sign flip, at 66.4%, $+0.049$ $[-0.002, +0.105]$. **The wide grid moves the comparison to the peak, and there it is null.** The peak sits at $\texttt{rel} = +0.20$, go rate 57.5%, $\Delta\kappa = +0.179$; three control settings now land within $3\,pp$ of that rate, and against the nearest — a shuffle at 58.2% with $\Delta\kappa = +0.145$ — the double difference is $+0.034$ $[-0.035, +0.108]$, $p = 0.33$. Against the strongest of the three, the isotropic draw at 59.6% with $+0.155$, it is $+0.024$. Two of the four controls beat not steering on their own after Holm ($+0.145$, $p = 0.0010$; $+0.155$, $p = 0.0008$), so a vector with no persona contrast reproduces the persona's decisions better than leaving the model alone, by an established amount. **The agreement gain is therefore not shown to be specific to the contrast.** The earlier $+0.058$ is not withdrawn — it is the same quantity at a setting the narrow grid could reach — but the peak comparison supersedes it, and it is the peak the raw $+0.179$ is quoted from.

Rate matching is weaker than it looks, once the response is not monotone. A single control can hold the same rate at two coefficients, one on each side of its own κ peak, and the two are not interchangeable as nulls. The isotropic draw has two settings within $3\,pp$ of the peak's rate: 59.6% at $\Delta\kappa = +0.155$ and 56.8% at -0.109. Pairing on rate alone therefore admits a double difference anywhere from $+0.024$ to $+0.288$, and pairing on *closest* rate — which is what an automated match does — can select the setting at which the control's policy has already come apart. We report the strongest control at matched rate, because what a null has to answer is whether *any* contrast-free vector reaches the effect, and among the choices available that is the conservative one.

One thing we do not claim. The controls were swept at non-negative coefficients only, so their highest rate is the unsteered 77.4%, while the real direction raises the rate to 93.2% at $\texttt{rel} = -0.70$. That is not evidence that a contrast-free vector cannot raise the rate; it is evidence that none was asked to. The asymmetry on the rate axis is untested and we do not read it as specificity.

The sign is carried, and that is not a rate effect. Applied backwards the direction moves decisions away from the persona rather than nowhere. At $\texttt{rel} = -1.00$ the rate has fallen only $4.8\,pp$ and a control matched to it gains $+0.067$, while the real direction loses -0.404: a paired difference of -0.471 $[-0.569, -0.369]$. A rate displacement of that size cannot produce it in either direction, so whatever the vector transports has an orientation.

Split by what the two prompts do. Of the 300 items, the optimizer and persona instructions decide 93 differently and 207 the same. At $\texttt{rel} = 0$ the steered policy *is* the optimizer prompt, so it takes 0% of the contested items and keeps 100% of the shared ones by construction, which is also a check on the harness. At the rate-matched coefficient it turns 30.1% $[21.2, 39.6]$ of the contested items to the persona's choice while keeping 98.1% $[96.0, 99.5]$ of the shared; the rate-matched control turns 25.8% $[17.0, 35.1]$ and keeps 95.2% $[92.1, 97.8]$. In counts on this item set that is about 28 contested decisions won for 4 shared lost, against the control's 24 for 10. Pushed further, to $\texttt{rel} = +0.40$, the real direction reaches 66.7% of the contested items but has given up 26.1% of the shared ones, which is the same trade the κ curve turns over on.

The coefficient did not calibrate out of fold. Choosing a coefficient by the go rate it produces on one half of the items and evaluating it on the other leaves landing errors of 14.7 to $34.7\,pp$ against the twelve cohorts. The cause is not the grid: the same $\texttt{rel}$ produces rates 18 to $28\,pp$ apart on the two halves, against $7.5\,pp$ between the halves unsteered, for directions agreeing at cosine $+0.9975$. **The rate a coefficient produces is not a property of the direction alone**, and the error is not explained by grid coarseness.

Reproduction, and what bf16 costs. Recomputing the argmax rates from the sampled run reproduces the sweep at ten of twelve settings and differs by two decisions of 81 at $+0.20$ and $+0.40$: bf16 inference with CPU offload is not bit-identical across runs and near-tied items flip. Against that, the two layers' independent runs return the unsteered setting identically to every printed digit on the same half — go rate 0.7407407, sensitivity -0.2098214, $\texttt{go_boost}$ -0.40344946 — so the non-determinism is confined to settings where the intervention is applied. One decision per setting is the granularity at which every figure in this appendix should be read.

Against a calibrated sweep. Blas et al. (2026) sweep the same family of intervention with a calibrated magnitude and an unbounded search and report steering effects that vary *approximately linearly* with the coefficient. Our response is not linear: it falls over eight of ten settings and turns back at both ends. Neither of the two obvious explanations survives: the direction is not inert, since the source prompts differ by $25.3\,pp$ and the coefficient moves the rate along that axis, and the grid is not too coarse, since the sweep covers three times the human range. What remains is a response that is monotone over most of its range and bounded at both ends, which their framing does not cover.

One number here is a warning about how such results are usually reported. Spearman over the full sweep is -0.53 ($p = 0.13$) while the response is monotone over eight of ten settings: a rank correlation computed across a turning point understates a real dial exactly as it overstates a single-peaked one. A monotonicity claim resting on a rank correlation over a coarse grid — including one we nearly made, since $\rho = +0.92$ describes this same sweep read at the wrong position — should be read with that in mind.

What the two prompts do. They do not decide identically. Pooled over all 300 items the persona instruction lowers the go rate by $25.3\,pp$ under argmax and $25.7\,pp$ sampled, and the two folds agree to $2.2\,pp$ (Table 4). The direction is therefore extracted from a contrast this measurement can see. One thing to carry across: the intervention is measured on a policy this paper does not otherwise report, since under the optimizer instruction the model goes 76.4% of the time generating and 77.4% reading the answer token here.

J.1 A CONTROL DIRECTION THAT CARRIES A ROLE

The four controls above are norm-matched and carry no role at all, which is the right null for one question and the wrong one for another. A shuffle of the real vector's coordinates cannot tell us whether the direction carries *which* practitioner was named, because it carries no practitioner. The contrast pairs that could are already in the instrument, differing from the coach arm in one noun: `A7_nonsense_token`, `A6_unrelated_occupation` and `A8_typical_coach` (Appendix G). Extracting a direction from each and sweeping it at the same layer, norm, folds and items asks whether the activation direction carries the ladder Section 3 measures in the prompt. `A8` is re-run rather than reused: Section 5's direction comes from `P2_typical_veteran`, which carries *veteran*, and `A6` and `A7` do not, so comparing them against `P2` would put back the confound the three-noun design removes.

A gate was fixed before running, because a sweep on a near-null direction re-measures the isotropic control at three hours a time: at least one of the two control arms must move go|opt by $5\,pp$ or more. The gate measures the three prompts unsteered, on fold a's held-out half, at the answer-token argmax read-out — the row-3 procedure of Table 4 (runs `steer_A8_a_gate`, `steer_A6_a_gate`, `steer_A7_a_gate`).

arm	instruction	go|opt	Δ vs P0	$\|v\|$
`P0_optimizer`	the optimizer objective	81.5%	—	—
`A6_unrelated_occupation`	a typical *baker*	10.8%	$-70.8\,pp$	28.89
`A7_nonsense_token`	a typical *zorbil*	43.1%	$-38.5\,pp$	25.71
`A8_typical_coach`	a typical *NFL head coach*	53.8%	$-27.7\,pp$	19.29
`P2_typical_veteran`	a typical *veteran* coach	56.9%	$-24.6\,pp$	19.55

Table 16: The gate: the same instruction strings on Qwen2.5-14B-Instruct, unsteered, on the 150 items fold a holds out, of which 65 have `go_boost` > 0. $\|v\|$ is the norm of the mean activation difference extracted from that arm against `P0` at layer 30. Both control arms pass the gate. These are point estimates on one half and no interval is fitted to them.

An unrelated occupation displaces more than the domain's own. Appendix G reports the same quantity on gpt-luna — the go rate where going is the higher-value option, which needs no threshold and so does not depend on the R2 gate — and there it falls 29.3, 23.2, 16.5, 11.0% along P0, A7, A6, A8: zorbil below baker below coach in conservative displacement, the order Section 3 reads as proximity to the domain. Here the ascending order is coach, zorbil, baker. The two baselines are far

apart, 29.3% against 81.5%, so the magnitudes are not comparable and we do not compare them; the ordering is, and `A8` falls from the largest displacement of the three to the smallest of the three point estimates.

How much of that order is established. Not the whole of it. Baker above zorbil holds in both models and Spearman between the two orderings is -0.5, not -1: what moved is the rung the claim rests on. The gate wrote the joint decisions — one row per item per arm, and the three runs agree on `P0` item by item, so they scored one common half — and Table 17 reports them with an exact McNemar test on the discordant pairs and a bootstrap resampled over source games. A pair is called separated only where $p < 0.01$ *and* that interval excludes zero; the rule was written into the analysis before its output was read, and three pairs are compared with p values left raw.

pair	b	c	difference	95% CI	McNemar p	
baker − coach	0	28	$-43.1\,pp$	$[-55.4, -30.8]$	7.5×10^{-9}	separated
baker − zorbil	0	21	$-32.3\,pp$	$[-44.6, -21.5]$	9.5×10^{-7}	separated
zorbil − coach	6	13	$-10.8\,pp$	$[-23.1, +1.5]$	0.167	not separated
coach − veteran coach	1	3	$-3.1\,pp$	$[-9.2, +3.1]$	0.625	not separated

Table 17: The three noun pairs and the modifier pair, on the 65 eligible items of fold a's held-out half. b is the count where the first arm goes and the second does not and c the reverse; the exact test uses only those two, which the marginal rates in Table 16 do not determine.

The enumeration this replaces predicted the outcome exactly. Every joint table consistent with the marginals gives baker against coach $p \leq 1.5 \times 10^{-5}$ and baker against zorbil $p \leq 5.1 \times 10^{-4}$; for zorbil against coach the two rates differ by seven decisions, so the fewest discordant pairs is seven and the smallest attainable p is $2/2^7 = 0.0156$, above the 0.01 fixed here. No joint table on these items could have separated that pair, and the observed one does not: $p = 0.167$. We did not relax the threshold on learning which pair it excludes. The last row is a second negative result worth stating: adding *veteran* to the coach instruction does not measurably change the policy, on this population or on all 150 items ($-6.0\,pp$, $p = 0.012$).

So what the gate establishes is that an unrelated occupation displaces the policy further than the domain's own and further than a nonsense token. The first contradicts an ordering by domain proximity on its own, since that ordering predicts the reverse. It does not establish that the coach rung is last of the three.

The ordering survives the other half and the other read-out. Two checks were available without new inference and both were run. Fold b's disjoint half, 81 eligible items, gives -67.9, -43.2 and $-29.6\,pp$ for baker, zorbil and coach against its own optimizer arm at 74.1% — the same order, with its unsteered baseline $7.5\,pp$ below fold a's, which is the scale on which two disjoint halves of this item set disagree. Reading the answer token by sampling at $T = 0.7$ rather than by argmax, over all 292 parsed items, gives -69.2, -41.1 and $-26.0\,pp$: the same order again, under a decision rule Table 4 separates from the first. That run also covers `A1_role_only`, the bare coach, which no steering run can reach because the sweep's arm pool does not hold the modifier ablation. It displaces $-21.2\,pp$, less than the coach and less than the veteran coach at $-23.6\,pp$. **The modifier keeps its direction where the nouns do not**: dropping *typical* costs displacement here and in Appendix G's gpt-luna (-8.0 against the coach's $-18.3\,pp$), while the noun order inverts between them. One number in that run does not close: it puts the optimizer-to-veteran contrast at $-23.6\,pp$ where Table 4 row 2 reports $-25.7\,pp$ on the same read-out, and $2.1\,pp$ on a sampled policy over 292 items is within what resampling could produce but is not shown to be, so we report the two rather than treat them as one.

What the gate does not license. Every arm here fails R2: `go_boost` coefficients $+0.278$ ($p = 0.33$) for `P0`, $+0.286$ ($p = 0.27$) for `A6`, $+0.505$ ($p = 0.064$) for `A7`, -0.078 ($p = 0.72$) for `A8` and -0.018 ($p = 0.93$) for `P2`, with margin sensitivity negative in all five. So none of these supports a threshold: none passes R2 in this evaluation, and we report rates. `A7` is the only arm that lands inside the cohorts' go|opt band of 24.6 to 47.4%, at 43.1% against 2024's 43.06% — a rate landing on a cohort's rate, not on its decision rule, and the same arm fails R2. The distinction Section 4 draws for the prompt applies here unchanged.

Both folds reproduce their own unsteered row, and the first check for that was wrong. Fold a's gate returns the fold-a, layer-30, `rel` $= 0$ row of Section 5's sweep to every printed digit — go|opt 0.815385 on 65 items, `go_boost` $+0.277631$, sensitivity $-9.4\,pp$, logit margin 8.22 — and fold b returns its own, 0.740741 on 81 items with `go_boost` -0.403449, on separate executions months later with a different GPU memory split. The halves are the same halves for a reason independent of those matches: the split is a permutation of the frozen item file under a fixed seed and extraction fraction, all runs record the same file hash, and the three gate runs agree on `P0` item by item, which an agreement of aggregates would not imply. **The automated check we ran compared both folds against fold a's row** and so reported every fold-b arm as a mismatch, on correct data. The unsteered policy belongs to the items a fold evaluates and the halves are disjoint, so one reference cannot serve both; the check now holds one per fold. Nothing was discarded and no number here changed, but the released logs contain that false alarm and this is what it was.

The sweep, and what it found. All three directions were extracted and swept on both folds at layer 30 on Section 5's grid, cross-fitted the same way, and compared against that section's `P2`-derived direction at matched go rate. **At the settings where the two directions reach the same rate on the same side of the response's turn, no difference is detected**, and that does not depend on which matching rule is applied. The rule was fixed in advance — among a control's settings that reach the real setting's rate on the same side, take the smallest absolute coefficient — and the released script implements a different one, the nearest rate over every setting regardless of side. Both are reported here, per arm, at the setting each rule selects:

control	rule	real rel	matched rel	excess κ	95% CI
coach `A8`	nearest	$+0.10$	$+0.10$	$+0.0055$	$[+0.0000, +0.0175]$
	prespecified	$+0.10$	$+0.10$	$+0.0055$	$[+0.0000, +0.0175]$
baker `A6`	nearest	-0.50	-1.00	-0.0017	$[-0.0502, +0.0458]$
	prespecified	-0.50	-0.30	-0.0084	$[-0.0550, +0.0355]$
zorbil `A7`	nearest	-0.50	-0.30	-0.0269	$[-0.0752, +0.0196]$
	prespecified	-0.50	-0.30	-0.0269	$[-0.0752, +0.0196]$

Every rate here is a go rate on the items where going is the higher-value option, over 146 of the 300 — the 65 and 81 of the two folds, which `steer_crossfit` pools into one table before any of this is computed, so the comparison is over both halves and not over one. The two rules select the same setting for two of the three arms and differ for the baker, where the excess moves from -0.0017 to -0.0084 and the verdict does not. The coach row's lower endpoint is $+0.0000$ to four decimals with a two-sided p of 0.73: that is not an interval on the edge of excluding zero but one whose replicates return exactly zero often, which is what two directions deciding the same items the same way at a $0.0\,pp$ rate gap looks like. Three directions extracted from role-conditioned prompts, modifier-matched and norm-matched, buy the same agreement as the registered one once the rate is held — which is the objection this experiment was built for, answered against a null that carries a role rather than against a shuffle.

The prespecified rule also refuses the comparison the other one forces. Of each arm's eleven non-zero settings it declines to match four, five and three respectively, printing the reason and the smallest achievable gap rather than reaching for a setting across the turn. The $+0.780$ excess this appendix reports below, the one the nearest-rate rule produces at the real direction's peak, is one of those refusals: on the coach direction's own side of the turn nothing comes within $4.1\,pp$ of 57.5%, so the prespecified rule reports no match there and the row does not exist. **Its tie-break clause is not hypothetical either.** The rule says that where two coefficients of a direction reach the target rate the smaller absolute one is taken, and at the coach direction that case occurs: `rel` $= -0.90$ and `rel` $= -0.10$ both reach 85.6%, the rate to be matched, and they differ by 0.296 of κ. The nearest-rate rule has no reason to prefer either and takes the first it finds, -0.90, which returns an excess of $+0.319$ $[+0.216, +0.420]$ — an interval well clear of zero, and the strongest apparent evidence of specificity anywhere in this experiment. Under the prespecified rule the same row is $+0.023$ $[+0.000, +0.053]$, $p = 0.10$. So the executed rule produces two positive results the fixed rule does not: this one, from an arbitrary choice between tied settings, and the $+0.780$ above, from a match across the turn. Neither survives the rule we wrote down first.

And where the intervals do exclude zero they go the wrong way for specificity: at `rel` $= -0.90$, -0.70 and -0.50 the coach direction agrees with `P2`'s decisions *more* than `P2`'s own direction does,

by 0.057, 0.056 and 0.033 of κ. One of those three survives a Holm correction over that arm's eight matched rows ($p = 0.0003$). We report it because it is the same measurement as the rest of the table and points away from the direction being specific, not toward it.

Three limits on reading that, all recorded rather than discovered afterwards. It is an absence of a detected difference and not an equivalence: low precision, a failure common to all four vectors, and an ill-suited reference produce the same absence, and we do not read it as the direction carrying a domain register rather than a name. Second, **the released script's default matching rule is not the one prespecified**, which is why the table above reports both. The script takes the smallest rate gap over every setting; at the real direction's peak ($\texttt{rel} = +0.20$, 57.5%) that is $\texttt{rel} = -1.00$ at 58.9% on the coach direction — the far negative branch, where that direction loses 0.601 of κ, and the excess comes out $+0.780$ $[+0.658, +0.897]$. A rate match taken across the turn is the artefact this appendix already warns about, and the prespecified rule does not make it: it reports no match at that setting. Both rules now run from one selector in the released script, which prints which rule produced a row and, where it declines, why. Third, at $\texttt{rel} = -1.20$ and -1.00 no role-derived direction moved the rate as far as the `P2` direction did, so those rows cannot separate the two explanations at all. What they do show is on the rate axis rather than the agreement axis: the registered direction reaches rates none of the three role-derived ones reach, which is a specificity claim about how far a vector moves the policy and not about which practitioner it reproduces.

K Related work, in full

Role-play prompting has been reported to improve zero-shot reasoning (Kong et al., 2024) and to leave task performance unchanged (Zheng et al., 2024). The negative findings are sharper: assigning a persona raises toxicity (Deshpande et al., 2023), persona-assigned models acquire implicit reasoning biases (Gupta et al., 2024), and role-play prompts hurt in identifiable conditions (Kim et al., 2024). That literature evaluates personas against a task standard, not against the distribution of practitioners the persona names, because in the domains studied no option-level value model prices the alternatives.

Persona vectors (Chen et al., 2025) extract a direction from contrastive persona prompts, as Section 5 does, and use it to monitor and steer how a trait *reads* in open text. What is at stake here is not how a trait reads but whether a policy depends on the value of the action, and the controls of Section 5 bear on that difference: a direction can move the rate and carry a sign without the agreement it buys being specific to the contrast it was extracted from.

The closest framing is CoMPosT (Cheng et al., 2023b), which defines caricature through **individuation**, whether the simulated output differs from the default, and **exaggeration**, whether it overshoots a meaningful response. Our slope and level are the same two components measured against a recorded human distribution in the domain's own units rather than scored on open-ended text. The two dissociate in our data, which the text-space framing does not predict. gpt-luna overshoots every cohort while misordering two of the five: its slope clears zero but its rank correlation does not reach significance, which is exaggeration with the cohorts only partly distinguished. gpt-4.1 recovers the order exactly, at a third of the correct scale, and still carries an offset. Scoring either on open-ended text would register a persona effect in both cases and separate neither. Related bounds come from stereotype content in generated personas (Cheng et al., 2023a) and from impersonation changing capability rather than style (Salewski et al., 2023).

A separate line already reports that persona prompting steers poorly toward intended target populations and flattens within-group heterogeneity (Santurkar et al., 2023; Argyle et al., 2023; Bisbee et al., 2024; Durmus et al., 2024), with benchmarks organised by question domain, steering method and distributional representation (Meister et al., 2024). Those targets are survey-response distributions, which have no natural unit; ours is a distribution with a domain-native unit and a counterfactual optimum, which is what lets us report a landing error rather than a divergence. Chang et al. (2026) argue separately that steerability evaluation needs a course correction. Cohort-targeted behaviour is achievable by other means: Maia (McIlroy-Young et al., 2020) trains a network per chess rating band and Maia-2 (Tang et al., 2024) replaces those with one network and a skill-level embedding. What our results add is that it is not achievable through the tested prompt interfaces in these two models, which do read the target.

Fourth-down decisions have served as a testbed since Romer (2006), with the counterfactual model from Baldwin (2026) over play-by-play from Carl and Baldwin (2026). Its limits are argued: Brill et al. (2025) urge humility about analytic recommendations, and Sandholtz et al. (2024) recover risk preferences rather than treating deviation as error. We inherit that debate rather than settling it.

L THE EXPLORATORY RESULTS AS PICTURES

Sections 2 and 3 report these results as tables, which is the compact form. The same three are drawn here, because the shape of each is the argument and a table asks the reader to reconstruct it. Figure 4 places the measured arms against the recorded range; Figure 5 shows which part of the instruction string carries the displacement; Figure 6 shows that the effect needs the domain rather than the decision structure.

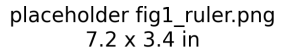

Figure 4: The ruler. The band is the range of season-cohort thresholds and the points are the cohorts with their intervals; the arms are placed against it with theirs, because a point estimate compared against a band carries more apparent precision than the arm has. $\tau = -\beta_0/\beta_1$ is a ratio, so an arm whose $\Pr[\text{go} \mid \text{margin}]$ is nearly flat has bootstrap replicates in the hundreds of points and no usefully bounded interval — drawn to scale those arms set the axis to $\pm 1000\,pp$ and reduce the $3.65\,pp$ band this figure exists to show to a line. The axis is therefore set from the record and the arm point estimates, and an interval leaving it is drawn to the edge with a caret. Eleven arms are so marked — every open-weight arm here but one, and no frontier one: Qwen2.5 at 1.5B, 3B and 7B on both conditions, 14B on both in each of its two variants, and 32B on the optimizer, with limits running out to $\pm 1300\,pp$. Ten of the eleven are also excluded from the axis calculation: nine because their intervals are wider than $40\,pp$ or unbounded, and one because a narrow interval is not the same as a nearby placement. Qwen2.5-32B's optimizer arm is estimated tightly, at a $13.6\,pp$ interval, and lands more than a ruler width below the recorded range; letting it set the scale would flatten the $3.65\,pp$ band this figure exists to show. **An arm is on the axis if it lands within one ruler width of the record**, and is clipped with its value printed otherwise, whichever of the two reasons applies. The band drawn here is the reweighted one the body reports, 2.45 to $6.10\,pp$. **This is not the same test as R2, and the two should not be read off one axis.** A wide interval here is a property of the ratio τ under a nearly flat univariate fit. R2 is a multivariable gate on `go_boost` with the situation covariates and errors clustered by item, and the two come apart in both directions: Section 2.2's worked example has a strong univariate correlation ($\rho = +0.53$) and a tight-looking τ, and fails R2. So the carets mark arms whose τ has no usable interval, not arms R2 rejects. The R2 verdict is marked on each arm rather than left to be looked up: a square is an arm whose τ the body can report, a cross one whose τ it withholds, and no τ from a crossed arm is quoted anywhere in the body (the per-arm fits are in Tables 12 and 14). Of the seventeen arms drawn, five pass and twelve fail, and that split is exactly the API-only against open-weight one: every frontier arm here passes and every open-weight arm fails. Seventeen and not eighteen because gpt-luna was not run on `P2_typical_veteran`, the registered persona prompt; its persona-side arms are the modifier ablation and the occupation set, in Appendices G and I. Section 2.

The two-cohort slope-and-offset figure that accompanied an earlier draft is not reproduced: it is defined for exactly two targets, and the analysis it illustrated is superseded by the five-cohort fit in Section 4.

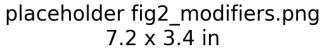

Figure 5: The modifier decomposition, two models against a shared baseline per model: an invented role, a bare role, the role with each modifier added, and the superlative in grey, from runs AB and BO. Every arm sits on the same side of its own optimizer in both models and they differ in size, not direction. **That is a statement about the bars and not about the modifiers**: the increments between bars do not all agree in sign, and Section 3 reports the one that does not — adding *veteran* to the bare role is worth $+0.28$ in one model and -0.20 in the other. The adjective-against-occupation comparison discussed in Section 3 is *not* plotted here — it is a different run (OC/OC2, whose baker arm has no bar in this figure) and is in Tables 6 and 8. Appendix G.

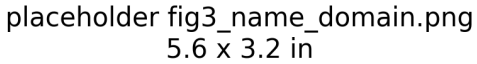

Figure 6: Name against domain: the four combinations of domain — the recorded one and a synthetic one with the same decision structure — and name, real or invented, each showing the conservative displacement from that domain's own optimizer arm — positive meaning more cautious, which is the negative of the change in go rate Table 3 prints for the same arms. Only claude-sonnet-5 and gpt-4.1 have an arm in all four cells; Qwen2.5-32B has the football real-name and synthetic invented-name cells only, so no within-model domain contrast can be read from it. The *other* open-weight models in this paper ran only the football side, so the contrast this figure is about is undefined for them and they are not shown. Section 3.1.