Decodable but Unused: What Language Models Represent (and Ignore) When Choosing Between Two Options

Anonymous ACL submission

Abstract

When a language model is asked to pick the better of two options under a stated preference, does it internally compute which option wins, and does that computation drive its answer? We study two-option consumer-choice prompts (health insurance and coffee) whose gold label is the argmax of an experimenter-defined utility over the stated attributes. In Pythia-160M and Pythia-1.4B, layerwise probes decode the utility-maximizing option from the residual stream near ceiling, far above random-label controls and surface baselines. Because a probe could reconstruct a linear utility on its own, we add a task with a nonlinear (XNOR) reward that no linear classifier on the raw attributes solves (the best is capped near 60–65%); probes on the activations of four models, including Qwen2.5-1.5B base and instruct, still reach 78.9–99.4%, stable across dataset seeds. Two results limit this. Probes do not transfer between demand regimes, and no model's emitted choice tracks the utility argmax. Qwen2.5-1.5B-Instruct gives a valid single letter on every prompt, yet picks the same letter whatever the options. Steering Pythia-160M along the decoded direction flips no choices and, against a null of 100 equal-norm random directions, moves the answer logits more than every random direction in only one of eight configurations, so the interventions we tested find no evidence that the representation is used for the choice. Near-ceiling decodability can coexist with no measurable behavioral use, and telling the two apart takes adversarial baselines and interventions in addition to control tasks.

1 Introduction

A growing line of work treats large language models as stand-in decision-makers: given a description of options and a stated preference, an LLM's answer is read off as if it were a human or agent choice (Horton, 2023). Whether this is a reasonable thing to do is partly a mechanistic question. If a model is asked to choose the better of two options under an explicit rule, does it *compute* which option wins internally, or does its answer come from something else, such as surface pattern-matching, positional habits, or a token-level prior unrelated to the stated rule? These are different claims with different consequences for using LLMs as economic simulators, and they are answerable with interpretability tools: linear probes (Alain and Bengio, 2016) for whether a quantity is linearly present in activations, control tasks (Hewitt and Liang, 2019) for whether a probe result reflects real structure or probe expressiveness, and causal interventions (Geiger et al., 2021) for whether a represented quantity affects behavior.

We study this with a minimal synthetic task: a customer choosing between two options (two health insurance plans, or two coffees) under an explicitly stated preference, with the gold label defined as the argmax of a deterministic utility over the attributes named in the prompt. Two demand regimes with opposite structure, **insurance** (inelastic: coverage decisive, price nearly irrelevant) and **coffee** (elastic: price decisive), let us ask whether any representation we find is specific to one regime or shared. We probe Pythia-160M and Pythia-1.4B (Biderman et al., 2023) layer by layer on both tasks below, and add Qwen2.5-1.5B-Instruct and its base model Qwen2.5-1.5B (Qwen Team, 2024) on the nonlinear task (four models in all). We ask four questions:

1. Is the utility-maximizing option linearly decodable from the residual stream at the final pre-answer token, beyond a random-label control and surface-statistics baselines, and is the continuous utility margin decodable as well as its sign?
2. Does a probe trained on one demand regime

transfer to the other, or is the representation regime-specific?

3. Does the model's own next-token choice track the decodable utility argmax?

4. Is the representation *causally* connected to the answer: does intervening on the probe's own direction move the answer logits more than a random direction of equal size, and does it change the choice?

A fifth question is methodological. Every utility above is linear in grid-normalized attributes, so in principle a linear probe could solve the task itself by reconstructing the argmax from linearly-encoded attribute values, without the model performing any comparison at all. To close this gap we build a second task whose reward is a nonlinear (XNOR) interaction between two attributes, verify that no linear classifier solves it exactly from the raw inputs, and re-run the full probing pipeline across four models (five dataset seeds for three of them). This task also permits a matched comparison of a base model with its own instruction-tuned counterpart, which the saturated linear task cannot resolve.

Contributions. (1) A controlled two-regime consumer-choice probing study showing near-ceiling, selective, but regime-specific decodability of a utility-maximizing choice that the models' own choices do not track; steering along the decoded direction changes no choice and, in all but one configuration, moves the answer logits no more than random directions of equal norm. (2) A nonlinear-utility variant that rules out the "probe reconstructs a linear function of the inputs" explanation, across four models and up to five seeds, finding the same qualitative picture. (3) A transfer check on prompts built from two real consumer-choice datasets, showing the synthetic-trained representation does not transfer even though the real choice is itself decodable from real-prompt activations.

2 Related Work

Probing and its confounds. Linear probes are widely used to ask whether a network's activations linearly encode a target quantity (Alain and Bengio, 2016), but a probe with enough capacity can fit structure that was never really "in" the representation (Belinkov, 2022). Hewitt and Liang (2019) propose control tasks (structureless labels over the same inputs) to calibrate how much of a probe's accuracy is attributable to probe expressiveness rather than encoding. We use this throughout. Control tasks rule out probe memorization, but they cannot rule out a probe that solves the task from linearly available inputs; for that we add an embedding-level restricted-value probe. A separate line asks whether decodable information is *used*: probes can recover properties a model does not need for its task (Ravichander et al., 2021), and amnesic probing (Elazar et al., 2021) and closed-form concept erasure (Belrose et al., 2023) test use by removing a property and measuring the behavioral change.

Emergent representations. A line of interpretability work finds that sequence models trained without explicit supervision for a task-relevant quantity nonetheless represent it linearly: board state in Othello-playing transformers (Li et al., 2023a; Nanda et al., 2023), and geographic and temporal coordinates in language models (Gurnee and Tegmark, 2024). Such structure recurs. We ask a companion question in the spirit of Belinkov (2022): when it is found, does the network that carries it use it? Causal-abstraction and intervention methods (Geiger et al., 2021) are the standard way to move from a correlational probing result to this stronger causal claim, and adding probe-derived directions to activations can change model outputs (Li et al., 2023b); our steering experiments apply the same logic, with a many-direction random null as the control.

LLMs as economic agents. Horton (2023) argues that LLMs given explicit preferences and budget-style scenarios reproduce qualitative patterns from behavioral economics, motivating their use as low-cost stand-ins for human subjects. Our results bear on a mechanistic precondition for that use: in the models we test, the stated preference and the attributes needed to act on it are both represented with high fidelity, but we find no working link from that representation to the emitted choice.

3 Task and Data

3.1 Linear-utility consumer choice

Each prompt describes a customer choosing between two options (A and B) with explicitly stated attributes and a stated preference, ending immediately before the answer token ("…*Which plan does the customer choose? Answer: Plan*"). Insurance options have a monthly premium ($180–

$480), an annual deductible ($500–$3,000), and a coverage percentage (70–95%); coffee options have a price ($2.00–$6.50), a quality rating (5–10), and a size (8–20 oz). One fixed template per regime; only attribute values vary. Complete prompts are given in Appendix A. The prompt states a qualitative priority (e.g., "values comprehensive coverage far more than a low price") and does not give the numeric weights below.

Each option's utility is a fixed linear function of its grid-normalized attributes: insurance (inelastic), $U = 3.0\,\mathrm{cov} - 0.6\,\mathrm{prem} - 0.9\,\mathrm{ded}$; coffee (elastic), $U = -3.0\,\mathrm{price} + 1.0\,\mathrm{qual} + 0.5\,\mathrm{size}$. In both regimes the decisive attribute's weight is double the combined weight of the others, mirroring the stated preference. The label is $\arg\max(U_A, U_B)$; pairs with $|U_A - U_B| < 0.25$ are rejected so labels are unambiguous, and exact-duplicate scenarios are rejected so no prompt spans both splits. Both options are sampled i.i.d. from the same grid, so the maximizer's position is exchangeable by construction; we assert a 40–60% choice share in every regime×split cell (realized: insurance 50.3%/49.5% train/test, coffee 50.8%/52.4%; maximizer at position A in 49.9%/48.9% of scenarios). We generate 4,000 scenarios per regime (3200 train / 800 test) from a single seed (20260611); generation is deterministic and unit-tested. Following Hewitt and Liang (2019), each scenario additionally receives a control label from a seeded hash of its full attribute tuple: structureless, 50/50, and impossible to transfer across splits since scenarios are unique.

3.2 A nonlinear robustness variant

Because the utility above is linear, a linear probe could in principle solve the task itself, reconstructing the argmax from linearly encoded attribute values without the model performing any comparison. We close this gap by replacing the utility with one that no linear classifier on the raw attributes solves exactly: each option's utility is a weak linear term in its three attributes (weights in Appendix A) plus a bonus $\gamma \cdot \mathbf{1}[x\text{-good} \Leftrightarrow y\text{-good}]$ for being *internally consistent* (jointly high or jointly modest on two threshold-gated attributes (x, y)), and no bonus when they mismatch. This XNOR structure is a checkerboard on the (x, y) grid: no linear classifier solves the task exactly, and the best raw-linear classifier is capped near 60–65% (a simpler AND-only interaction is not enough: a hyperplane recovers up to 76.5% test accuracy on it, so we discarded that design). Grid search under acceptance constraints (label balance 45–55%, a linear regression on the six raw attributes capped at $\leq$65% test accuracy, an interaction-feature random-forest oracle at $\geq$90%, verified stable across 12 resamples) fixes task parameters per regime (insurance: x=coverage$\geq$80%, y=deductible$\leq$ $1000; coffee: x=quality$\geq$9, y=size$\geq$16; γ=1.5 in both). The model never sees γ or the linear weights; the prompt states the thresholds and the consistency rule in natural language ("the customer strongly rewards a plan that feels internally consistent . . . "; Appendix A). We generate 4,000 scenarios per regime under the same balance/duplication/margin safeguards as §3.1, across five seeds (one for the Qwen2.5-1.5B base model).

4 Method

We extract `resid_post` activations with TransformerLens (Nanda and Bloom, 2022) at the final pre-answer token of each prompt, whose next-token prediction is the answer (extraction asserts, per example, that the indexed token is the expected final word). For each model, regime, and layer ℓ (0 = token embeddings, $1..L$ = post-block residuals) we fit an L2-regularized logistic probe (C=1, standardized features; scikit-learn, Pedregosa et al., 2011) on 3,200 training examples and report test accuracy on 800 held-out examples with 95% Wilson binomial confidence intervals. *Selectivity* (Hewitt and Liang, 2019) is main minus control accuracy at the same layer. Baselines: majority class; bag-of-words (logistic regression on raw token counts; since both options draw from identical grids, token identity alone cannot bind a value to an option); and mean-pooled layer-0 embeddings.

Transfer and margin. Each probe (scaler + classifier) fit on one regime's train split is evaluated unchanged on the other regime's test split, both directions, per layer. Ridge regression (α=10) from the same activations onto the continuous latent utility difference $U_B - U_A$ reports held-out R^2.

Behavior. At the same final position we compare the model's next-token logits for " A" vs. " B" and measure agreement with the utility argmax. Probing measures decodability, and this is a separate test of use (Belinkov, 2022).

Steering. At each regime's best-selectivity

layer we take the probe's raw-space unit direction $\hat{d}$ and, on 300 held-out prompts, either project it out of the final-token residual (a rank-one, single-site analogue of amnesic probing, Elazar et al., 2021) or add $s\,\sigma_{\text{proj}}\hat{d}$ with sign flipped against the true label ($s \in \{1, 2, 4, 8, 16\}$), measuring the change in answer logit difference and choice flips against a matched-norm random direction (Geiger et al., 2021). We repeat this with a class-conditional difference-of-means direction (argmax-A minus argmax-B final-token means) to check the null is not an artifact of the raw probe direction's small amplitude.

Nonlinear task diagnostics. For §5.5 we add two baselines specific to ruling out an input-side shortcut: *raw-linear* (logistic regression on the six stated numeric attributes) and *restricted value-token*, a linear probe on the model's own layer-0 (pre-computation) embedding of the six stated value tokens, mean-pooled per attribute: the model's own token geometry before any transformer block has run.

5 Results

5.1 The linear task is decodable from block one

Figure 1 shows the main result. At the embedding layer, probe accuracy is at chance (49.5%–52.4% across cells), as it must be: the final token is identical across prompts, so any task information there has to arrive through attention. By the output of the *first* transformer block, probes already reach 99.2% (160M, insurance), 100.0% (160M, coffee), 100.0% and 100.0% (1.4B), and remain near ceiling through the rest of the network (peaks: 100.0%/100.0%/100.0%/100.0%; Table 1). Control probes never exceed 51.5%–52.2% in any cell (peak selectivity 48.5–52.6 points). Bag-of-words (insurance 50.6%, coffee 52.2%), mean-pooled embeddings (49.8%–50.6%), and majority class (49.5%/52.4%) are all at chance, so surface statistics alone cannot solve the task: identical value grids for both options make token identity uninformative about which option a value belongs to. Ridge probes recover the *continuous* margin $U_B - U_A$ almost perfectly from layer 1 onward (test $R^2 \geq 0.988$ everywhere, with peaks rounding to 1.000), so the full latent index of the discrete-choice model is linearly present along with its sign.

Model	Regime	Peak acc. (95% CI)	Layer	Select.
160M	insur.	100.0 (99.5–100.0)	2	52.6
160M	coffee	100.0 (99.5–100.0)	1	48.5
1.4B	insur.	100.0 (99.5–100.0)	1	51.6
1.4B	coffee	100.0 (99.5–100.0)	1	49.0

Table 1: Peak probe test accuracy (%, n=800), best layer, and selectivity (percentage points above the same-layer control) on the linear task.

5.2 No transfer across regimes

Decodability is statistically indistinguishable between insurance and coffee at both scales: curves saturate at the same depth and CIs of every layer-$\geq$1 probe overlap across regimes. Yet the representations are not shared. A probe trained on one regime and evaluated on the other collapses to majority-class prediction at nearly every layer (best transfer: 52.4%/50.5%, 160M; 52.4%/64.9%, 1.4B – the single exception, 1.4B coffee→insurance at layer 10, is isolated to that one layer). The utility-argmax information lives in regime-specific directions, and these probes find no prompt-general "maximizer" axis.

5.3 Behavior does not track the decodable argmax

The model's emitted choice does not track the experimenter-defined utility argmax, despite that argmax being decodable from its activations (Table 2). Agreement is at chance in all four cells, and the mechanism is a deterministic, content-independent answer: Pythia-160M emits " A" on 100% of test scenarios in both regimes; Pythia-1.4B emits " B" on 100%. The prompt gives no numeric weights, but its stated priority is enough to recover the label: following it (higher coverage; cheaper coffee; ties broken on the remaining attributes) agrees with the gold label on 96.1%/94.4% of test scenarios (Appendix A). A label-swap control, which re-renders each scenario with the two option letters exchanged and the descriptions in their original order, shows the fixed answer is *position* bias in some cells and *answer-token* bias in others (Table 3), never tracking the utility comparison either way. In Figure 1 (dash-dotted line), behavior sits flat at chance beneath a probe curve at ceiling from block 1. Few-shot prompting does not change this: with up to 8 balanced in-context exemplars (correct answers shown, answer letters counterbalanced), agreement never exceeds 52.6%/52.4% across $k \in$

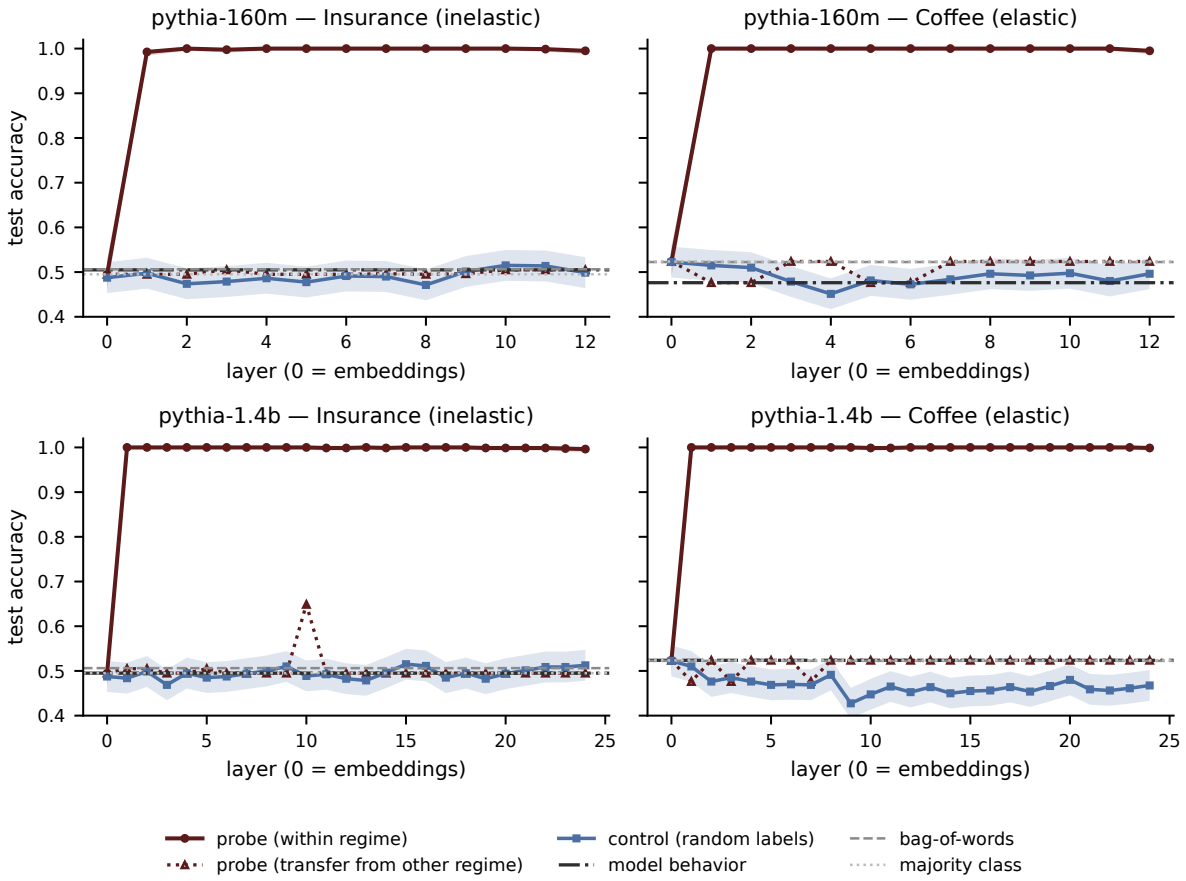


Figure 1: Layerwise test accuracy of logistic probes decoding the utility-maximizing option from `resid_post` at the final pre-answer token (oxblood: within-regime, with 95% Wilson CIs; oxblood dotted: the same probe transferred from the other regime; steel blue: Hewitt–Liang random-label control). Dash-dotted: the model's own behavioral accuracy at the same position; dashed/dotted gray: bag-of-words and majority-class baselines. Rows: Pythia-160M, Pythia-1.4B; columns: insurance (inelastic) and coffee (elastic).

Model	Regime	Behav. agree. (95% CI)	Chose B
160M	insur.	50.5 (47.0–54.0)	0.0
160M	coffee	47.6 (44.2–51.1)	0.0
1.4B	insur.	49.5 (46.0–53.0)	100.0
1.4B	coffee	52.4 (48.9–55.8)	100.0

Table 2: Agreement (%) of the model's next-token choice with the utility argmax on the test split (n=800), and the share (%) of scenarios where the model chose option B.

Model	Regime	Fixed-answer mechanism
160M	insurance	position (first option)
160M	coffee	answer token (" A")
1.4B	insurance	answer token (" B")
1.4B	coffee	position (second option)

Table 3: Label-swap control: each model's fixed choice is driven by option *position* in one regime and by the answer *token's identity* in the other, so no single mechanism covers all four cells. In every cell the choice is independent of the utility argmax.

$\{2, 4, 8\}$ and both regimes, so the pattern persists with in-context examples.

5.4 Steering: no choice flips, little beyond chance

If the represented comparison fed the answer, intervening on it should move the answer logits (Geiger et al., 2021). We steer Pythia-160M only (pairs are insurance/coffee). At its best-selectivity layers (8/4), the probe direction is a low-amplitude axis (σ_{proj}=0.012/0.006 against final-token residual norms of 28/15). At $16\sigma_{\text{proj}}$ the mean shift in the answer logit difference is $-0.023/-0.008$, i.e. slightly *against* the pushed option. Bootstrap 95% CIs over the 300 prompts are narrow (half-width $\leq$0.00187 in every condition; Appendix C), but they only reflect prompt

sampling. The informative control is a null of 100 random directions of equal norm, whose shifts span $[-0.018, 0.015]/[-0.012, 0.012]$ (central 95%). The insurance shift exceeds all 100; the coffee shift is matched or exceeded by 19% of them. Projecting the direction out moves the logit difference by $0.080/-0.087$, which 9%/5% of random ablations match. A class-conditional difference-of-means direction, with projection std up to an order of magnitude larger, shifts the logits in the same (negative) direction in all 6 layer×regime cells, but inside the random null's central 95% in every one. Against baseline logit gaps of $-0.59/-1.98$, no intervention at any scale flips a single choice. Steering changes no choices, and apart from the insurance probe direction its logit shifts fall within the range of random perturbations of the same size.

Model	Regime	Raw	L0	Peak	Behav.
160M	insur.	62.5	65.6	78.9	47.1
160M	coffee	59.2	60.4	82.4	50.6
1.4B	insur.	62.5	65.6	93.1	52.9
1.4B	coffee	59.2	60.5	99.4	52.0
Qwen-I	insur.	62.5	63.7	88.0	53.4
Qwen-I	coffee	59.2	60.5	97.0	48.2
Qwen-B	insur.	62.5	63.9	90.9	47.1
Qwen-B	coffee	59.2	60.5	98.0	50.2

Table 4: Nonlinear task (%, n=800): raw-scalar linear baseline, restricted embedding-level (layer-0) probe, peak full-token probe, and behavioral agreement with the utility argmax. Qwen-I = Qwen2.5-1.5B-Instruct; Qwen-B = its base (non-instruction-tuned) counterpart, Qwen2.5-1.5B. The two Pythia models share a tokenizer in which every stated insurance value is one token, so at layer 0 this probe is equivalent to a linear probe on value identity; hence their identical insurance L0 (a one-hot value probe also gives 65.6).

5.5 Robustness: a nonlinear choice rule

Two baselines rule out an input-side shortcut on the harder task (pairs below are insurance/coffee). *Raw-linear* is, by design, ≤65% (62.5%/59.2%). *Restricted value-token*, the model's own precomputation embedding of the six stated values, is similarly capped (65.6%/60.4%, 160M; 63.7%/60.5%, Qwen-Instruct). A nonlinear oracle (an interaction-feature random forest) confirms the task is solvable (99.2%/99.4%). Against these, the full final-token probe reaches 78.9% (160M), 93.1% (1.4B), and 88.0% (Qwen-Instruct) on insurance, and 82.4%/99.4%/97.0% on coffee, climbing gradually with depth (Figure 2), with label-shuffled controls pinned at chance throughout (50.5%–53.1% at the peak layer, every model, every regime). The representation gain over the embedding-level baseline is 13.2–27.5 points on insurance and 22.0–38.9 points on coffee. A linear read of the raw or embedded inputs cannot produce this gain, so it reflects computation inside the model. Five dataset seeds for every model except Qwen-B confirm these numbers are stable (peak accuracy moves ≤1.3 points from its single-seed value; 97.2% ± 0.28 pp on Qwen/coffee).

Behavior again does not track the utility argmax under the harder task: agreement with the nonlinear-utility argmax is at chance in all eight model×regime cells (Table 4), and each model again answers with a fixed, largely content-independent letter, although applying the stated rule alone (bonus first, then the cheaper option) would agree with the gold label on 85.4%/79.6% of insurance/coffee test pairs. We checked prompt formatting for Qwen2.5-1.5B-Instruct. It was prompted through its chat template with a natural-language description of the rule (never the symbolic formula) and asked for a bare " A"/" B" answer, and it gave a valid single-letter response on 100% of 500 held-out scenarios in both regimes. A label-swap check (as in Table 3) shows pure answer-token bias: token stability 1.000 and position stability 0.000 in both regimes.

Two results are new relative to the linear task. First, the representation is often *partial*: peak accuracy trails the oracle by 1.4–20.3 points in seven of eight cells, most for Pythia-160M, whereas the linear task saturates. The exception is Pythia-1.4B on coffee, which matches its own oracle almost exactly (99.4% vs. 99.4%). Second, Qwen-B's peak accuracy is slightly *higher* than Qwen-I's in both regimes (insurance: 90.9% vs. 88.0%; coffee: 98.0% vs. 97.0%), and Pythia-1.4B beats both; §6 discusses how far this single matched pair goes.

5.6 Transfer to real-choice prompts

We test whether the representation and the behavioral pattern survive transfer to prompts built from real human choices. We map two public datasets to our regimes: Train's stated-preference electricity-supplier data (Huber and Train, 2001) (inelastic) and scanner-panel yogurt brand choice (Jain et al., 1994) (elastic). Each is reduced to binary (chosen vs. a non-chosen alternative) pairs with counterbalanced position (446/1171 pairs), rendered with the matching synthetic tem-

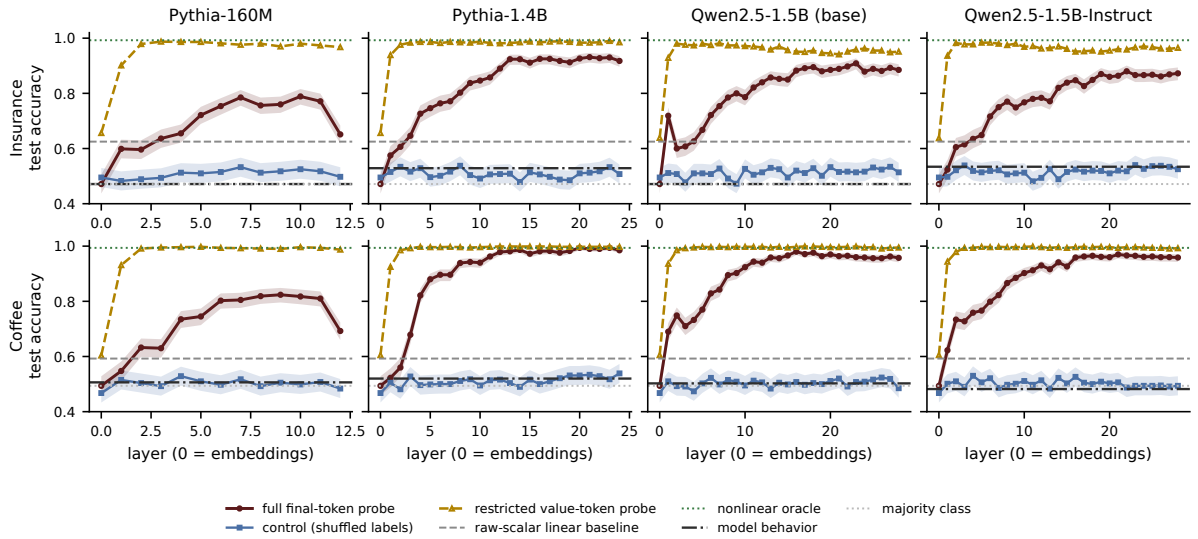


Figure 2: Layerwise probe accuracy on the nonlinear (XNOR-interaction) task for four models (columns) and both regimes (rows). The full final-token probe (oxblood) climbs with depth, well above the raw-scalar linear baseline (gray dashed) and the restricted value-token probe's layer-0 (pre-computation) value (amber). Amber points at layers ≥ 1 read the value tokens' computed states and are not an input-side baseline. Label-shuffled control (steel blue) and model behavior (black dash-dot) stay at chance; the nonlinear oracle (green dotted) upper-bounds achievable accuracy.

plate, and labelled two ways: the human pick, and the argmax of a conditional logit fit to the full data. The synthetic-trained probe does *not* transfer: for Pythia-160M, best layerwise accuracy is 52.5%/51.8% against the human label and 50.0%/51.8% against the utility argmax, at the majority baseline (50.5%/49.3%; Pythia-1.4B is at most 56.1%, best of 25 layers). Yet the choice *is* decodable within real data: a probe trained on real activations (group-split by household) reaches 69.2%/73.8% for the human pick and 90.7%/96.1% for the utility argmax. The synthetic-trained probe does not carry over, as with the regime-specific encoding found synthetically. Behavior again does not track either label on real prompts: Pythia-160M and Pythia-1.4B each emit a single fixed answer token and agree with neither the human pick nor the utility argmax above chance. Human picks themselves diverge from the fitted utility argmax on 28.9%/23.5% of pairs, and this choice noise caps how closely any utility-tracking model could agree.

6 Discussion

What near-ceiling probes do and do not show. Because our main utilities are linear in grid-normalized attributes, a logistic probe could in principle compute the argmax itself from activations that encode the six values linearly. On the linear task a restricted probe shows that this happens: reading only the six stated value tokens decodes the argmax and the exact margin at 100.0%/R^2=1.000, and at 100.0%/R^2=1.000 already at the embedding layer, where the full final-token probe is at chance (49.5%). On the linear task, then, the comparison outcome is linearly available from the encoded values without any model-side computation. §5.5 was built to close this confound. With a rule no linear classifier solves exactly, the qualitative picture is unchanged and now rests on computation inside the model, though decodability is graded rather than saturated.

Together with transfer and steering, the results separate three claims: (i) the information is *linearly present* beyond a random-label control (both tasks, all models); (ii) that presence generally requires model-side computation rather than a linear read of stated or embedded inputs (task 2; task 1's restricted probe tells us when it does *not*); and (iii) the information does not determine the emitted choice: behavior is at chance, and steering flips no choices while moving the logits little more than random directions do. Treating (i) alone as evidence that the model "computes utility" conflates all three. Our main methodological suggestion is

to test them separately on any task with a combinatorially simple ground truth.

Base versus instruction-tuned. The nonlinear task admits one matched comparison: Qwen2.5-1.5B-Instruct's derived representation is slightly weaker than its own base model's on identical data (88.0% vs. 90.9% on insurance, 97.0% vs. 98.0% on coffee), and both trail Pythia-1.4B (93.1%/99.4%), a different model family at similar scale. At this scale, then, instruction-tuning did not strengthen the representation. One single-seed pair at one scale cannot show that representation strength generally tracks scale more than instruction-tuning; that question needs several base/instruct pairs across scales.

Why saturation at block 1 is plausible (linear task). With a margin of $|\Delta U| \geq 0.25$ and a rigid template, one round of attention can gather the value tokens into the final position. The nonlinear task's gradual climb fits this: an interaction term must combine two attribute spans, and pooling activations per attribute span shows the span that cannot yet attend to the others stays at chance at every layer while later spans climb.

For LLM-as-economic-agent work. Horton (2023)'s caution can be made more mechanistic. In the models we test, the stated attributes are well represented: ridge probes recover the exact latent utility margin, and under a nonlinear rule the model builds a partial but substantial derived representation. Our interventions find no clear link from that information to the emitted choice. Whether larger or differently-trained models develop a transferable maximizer direction that interventions can use to change choices is the natural next empirical question.

7 Conclusion

In a minimal synthetic consumer-choice setting, two Pythia models linearly and selectively represent the utility-maximizing option (and, under a linear utility, the exact continuous margin) almost perfectly from the first transformer block. The encoding is regime-specific (no cross-regime transfer), not reflected in behavior (every model answers with a fixed, content-independent letter), and not shown to be used (steering flips no choices and moves the logits little more than random directions of equal norm). Replacing the linear utility with a rule no linear classifier solves exactly reproduces this picture from computation inside the model rather than a linear read of the inputs, across four models and up to five seeds. For LLM-based economic simulation, the bottleneck in the models we test is the link from representation to choice.

Limitations

All models are small (160M–1.5B parameters) and, on these rigid prompts, emit a near-constant answer, so agreement alone cannot reveal partial use. The prompts state qualitative priorities and thresholds, not the numeric weights (Appendix A). Each task uses one template per regime, and the nonlinear task one interaction form. Steering (Pythia-160M) intervenes at one layer and one position, up to $16\sigma_{\text{proj}}$; stronger, multi-site, or patching-style interventions could find choice-changing coupling we did not. The base/instruct comparison is one single-seed pair at one scale, and the real-choice test reduces multi-alternative occasions to binary pairs (details in Appendix B).

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations (ICLR)*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743. Association for Computational Linguistics.

John J. Horton. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? Working Paper 31122, National Bureau of Economic Research.

Joel Huber and Kenneth Train. 2001. On the similarity of classical and bayesian estimates of individual mean partworths. *Marketing Letters*, 12(3):259–269.

Dipak C. Jain, Naufel J. Vilcassim, and Pradeep K. Chintagunta. 1994. A random-coefficients logit brand-choice model applied to panel data. *Journal of Business & Economic Statistics*, 12(3):317–328.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377. Association for Computational Linguistics.

A Exact Prompts and What the Model Is Told

Pythia and Qwen2.5-1.5B (base) receive the raw prompt as a text continuation and are read at its final token (" Plan"/" Option"). Below are two complete test-split prompts, reproduced verbatim; only the attribute values vary across scenarios within a template.

Linear task, insurance (`insurance-0004`; gold: A, U_A=0.72, U_B=−0.24):

> A customer is choosing between two health insurance plans. Plan A has a monthly premium of $450, an annual deductible of $2000, and covers 85% of medical costs. Plan B has a monthly premium of $210, an annual deductible of $1000, and covers 70% of medical costs. The customer values comprehensive coverage far more than a low price. Which plan does the customer choose? Answer: Plan

Nonlinear task, coffee (seed 20260611, item `0058`; gold: A, U_A=2.27, U_B=0.41; A earns the consistency bonus, B does not):

> A customer is choosing between two coffees at a cafe. Option A is a 20 oz cup with a quality rating of 10 out of 10, priced at $5.00. Option B is a 12 oz cup with a quality rating of 10 out of 10, priced at $2.50. The customer generally prefers a cheaper coffee. They also strongly reward a coffee that feels INTERNALLY CONSISTENT: either a modest coffee (quality below 9 out of 10 AND smaller than 16 oz) or an excellent coffee (quality at least 9 out of 10 AND at least 16 oz) earns a bonus in the customer's eyes. A MISMATCHED coffee -- high quality but small, or large but low quality -- earns no such bonus, no matter how good either attribute is on its own. Which option does the customer choose? Answer: Option

The other two templates. The linear coffee prompt uses the coffee attribute sentences above with the preference sentence "The customer is highly price-sensitive and mainly wants a cheap drink." and the same final question. The nonlinear insurance prompt uses the insurance attribute sentences above, followed by: "The customer generally prefers a lower premium. They also strongly reward a plan that feels INTERNALLY CONSISTENT: either a clearly COMPREHENSIVE plan (covers at least 80% of costs AND has an annual deductible of at most $1000) or a clearly BASIC

plan (covers less than 80% of costs AND has an annual deductible above $1000) earns a bonus in the customer's eyes. A MISMATCHED plan -- high coverage paired with a high deductible, or low coverage paired with a low deductible -- earns no such bonus, no matter how good either number looks on its own. Which plan does the customer choose? Answer: Plan"

Qwen2.5-1.5B-Instruct. The same text, with the final question replaced by "Which plan [option] does the customer choose -- A or B? Reply with just the letter, A or B, and nothing else.", passed as a single user turn through the model's chat template with the generation prompt appended; activations and the A/B logits are read at the last token of the assistant-turn opener.

What is and is not stated. The prompts give every attribute value and a qualitative preference; for the nonlinear task they also give both thresholds and the consistency rule. They never give the numeric utility weights, the grid normalization, or γ. For completeness, the nonlinear utilities on grid-normalized attributes are insurance: $-0.2\,\mathrm{prem} + 0.2\,\mathrm{cov} + 0.4\,\mathrm{ded} + 1.5\,\mathrm{bonus}$; coffee: $-0.2\,\mathrm{price} + 0.2\,\mathrm{qual} + 0.7\,\mathrm{size} + 1.5\,\mathrm{bonus}$, with pairs closer than $|\Delta U|{=}0.10$ rejected. The unstated linear terms are not always aligned with the prompt's wording (the insurance term mildly rewards a *higher* deductible; coffee size outweighs price), but they matter only when both options have the same bonus status: γ exceeds the largest possible difference in the linear terms, so whenever exactly one option earns the stated bonus, that option is the gold answer.

How much of the label the stated preference determines. To check whether behavioral failure could come from an underspecified instruction, we score simple rules that use only what the prompt states against the gold labels ($n{=}800$ test scenarios per cell). *Linear task:* picking the higher-coverage plan (ties: lower premium, then lower deductible) agrees on 96.1%, and picking the cheaper coffee (ties: higher quality, then larger size) on 94.4% (86.9%/85.8% without any tie-breaking). *Nonlinear task:* options differ in bonus status in 59.0%/55.6% (insurance/coffee) of test pairs, where the stated rule alone fixes the label; "bonus first, then cheaper" agrees on 85.4%/79.6% overall (84.5–88.2% and 78.4–80.9% across the five seeds). Every model's agreement is at chance (§5.3, §5.5), far below what the stated preference alone supports. We therefore describe the finding as an emitted choice that does not track the experimenter-defined utility argmax, and we do not claim the model discards a decision it was told how to make.

B Extended Limitations

Scale and behavior. All four models have 160M–1.5B parameters, and on these rigid prompts each emits a near-constant answer. With a constant answer, behavioral agreement cannot distinguish "no use" from "weak use swamped by an answer bias". The steering results (§5.4) bound the size of any such coupling, for the interventions tested.

Prompts. One fixed template per regime in each task (no lexical or paraphrase variation). The prompts state qualitative priorities and thresholds but not numeric weights (Appendix A); the rules there show the stated preference determines most, but not all, gold labels.

Measurement. The linear task's utilities are linear and its probe accuracies sit at ceiling, compressing between-cell differences; behavioral readout is a single greedy logit comparison, not sampled choice behavior.

Interventions. Steering intervenes at one layer and one token position (160M only), at scales up to $16\sigma_{\mathrm{proj}}$; stronger, multi-site, or patching-style interventions could in principle find coupling that changes choices where ours did not. We claim only that, under the interventions tested, the decoded direction changes no choice and, except the insurance probe direction, shifts the logits no more than random directions of equal norm. This bounds a weak coupling; it does not exclude one.

Nonlinear task. One interaction form (an XNOR consistency bonus; other nonlinearities untested). Five seeds confirm the numbers for Pythia-160M, Pythia-1.4B, and Qwen2.5-1.5B-Instruct; the base-model run is single-seed. The base/instruct comparison is one same-architecture, same-scale pair on identical data. That design isolates instruction-tuning, but it covers one model family at one scale (1.5B) and does not support a general claim about scale versus instruction-tuning.

Real-choice prompts. The transfer test (§5.6) reduces 4-alternative occasions to binary pairs; Electricity is stated-preference while Yogurt is revealed-preference (a confound in its own right); its pairs are panel data (households repeat), so real-data intervals are not independent; and real

Regime	Dir.	Layer	σ_{proj}	Steered	Random	Random null, 95% range	% null $\geq$	Ablation	CI $\pm$
Insurance	probe	8	0.012	-0.0227	$+0.0030$	$[-0.0181, +0.0148]$	0	$+0.080$	0.0002
Coffee	probe	4	0.006	-0.0077	-0.0024	$[-0.0120, +0.0117]$	19	-0.087	0.0002
Insurance	DoM	2	0.017	-0.0058	-0.0024	$[-0.0145, +0.0164]$	49	-0.023	0.0003
Insurance	DoM	8	0.163	-0.1073	$+0.0447$	$[-0.3507, +0.2574]$	47	-0.026	0.0019
Insurance	DoM	12	0.215	-0.1747	-0.0409	$[-0.2850, +0.2983]$	20	-7.922	0.0011
Coffee	DoM	2	0.027	-0.0316	-0.0181	$[-0.0548, +0.0616]$	22	-0.186	0.0011
Coffee	DoM	4	0.023	-0.0339	-0.0097	$[-0.0415, +0.0429]$	13	-0.140	0.0007
Coffee	DoM	12	0.104	-0.0994	-0.0191	$[-0.1389, +0.1115]$	13	-0.566	0.0007

Table 5: Pythia-160M steering at $16\sigma_{\text{proj}}$, n=300 held-out prompts. *Steered*/*Random*: mean shift in logit(B)$-$logit(A), signed positive toward the pushed option, for the steered direction and the original matched-norm random control. *Random null*: central 95% of the same mean shift over 100 fresh random directions of equal norm; *% null* $\geq$: share of them with an absolute shift at least as large as the steered one. *Ablation*: mean change in the logit difference from projecting the steered direction out. *CI* $\pm$: largest bootstrap 95% CI half-width (10,000 resamples over prompts) among the row's four means. DoM = difference-of-means direction. No condition flips a choice.

attributes are not on a clean grid. It is a transfer test on real-choice prompts and does not establish external validity in the usual sense.

Seeds. Linear-task sampling, extraction, and analysis derive from a single seed.

C Steering Uncertainty

Table 5 gives the full numbers behind §5.4. All interventions are deterministic forward passes, so the bootstrap intervals only reflect which prompts were sampled. They are narrow because the prompts share one template and each shifts by nearly the same amount. The random-direction null is the more informative comparison: it asks whether the steered direction moves the logits more than a generic perturbation of the same size. We re-derived every stored mean before computing the intervals, and all match the published runs. Against this null, only the insurance probe direction stands out: its shift is larger than all 100 random shifts, though only just (largest random: 0.022). With 100 draws that is an empirical p of about 0.01, which does not survive a Bonferroni correction across the eight configurations. A single random control (the *Random* column) falls near the middle of each null, so ratios against it overstate how unusual the steered shifts are. Projecting the difference-of-means direction out at the last block moves the insurance logit difference by about -7.9, toward the answer the model already gives on every prompt, so again no choice flips. This ablation removes the direction's large shared component along with its label-dependent part.