THE ORIGINS OF PRICE SENSITIVITY IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Price sensitivity is where an LLM's numeracy becomes an economic preference: acting on a price means representing it, comparing it, and deciding how much it counts against everything else. Models differ in their price sensitivity, which shapes the purchase advice they give; little research has investigated how training produces those preferences. OLMo 2 7B's public checkpoints and training data let us follow representation, comparison and weighting through pretraining, supervised fine-tuning, and direct preference optimization (DPO). Our central finding: accuracy in comparing prices does not reveal how strongly price influences a choice. Internally, magnitude sits closer to a logarithmic scale than a linear one, the same shape the price response takes. By the earliest checkpoint probed, the model already distinguishes numbers of equal length, which digit count alone cannot. A lower-price preference develops during pretraining, alongside improving comparison accuracy and before preference optimization. The training text offers a plausible source: an estimated third of genuine detected prices in sampled web text occur in choice-resolving comparisons. In the plain prompt, annealing and supervised fine-tuning reduce price weight relative to decisiveness about non-price attributes. DPO partly reverses this decline; its effect is smaller in the chat template and in four other models we checked. Separating numerical knowledge from its influence on decisions gives a framework for auditing purchasing agents' economic preferences.

1 INTRODUCTION

Price is one of the most fundamental levers in any market. Economics has studied for half a century how people trade it against other attributes: discrete-choice models infer preferences from the options people pick (McFadden, 1974), and conjoint experiments measure trade-offs by design (Green & Rao, 1971; Louviere & Woodworth, 1983). Our understanding of the same preferences in language models is thin by comparison. AI systems are beginning to make purchase decisions and recommendations on behalf of users, and agents can differ in how they trade price against quality, in strength and in shape (Allouah et al., 2026; Cherep et al., 2026; Kireyev, 2026). How does a price preference form during training, and how does later training change it? Does the model's response to price reflect a logarithmic internal number scale, like the one proposed for human number perception (Dehaene, 2003) and reported in released LLMs (AlQuabeh et al., 2025)? This paper develops a framework for auditing an LLM's price sensitivity and traces where in training that sensitivity comes from.

The question is now practical. Checkout protocols, card networks and agent wallets let an agent search, recommend and pay on a user's behalf (Stripe, 2025; Shopify, 2026; Visa, 2025; Mastercard, 2025; Circle Internet Group, 2026). Regulators are responding. The UK Competition and Markets Authority's 2026 review of agentic AI warns that consumers may be "steered towards products and services that are more profitable but less suited to their needs" (Competition and Markets Authority, 2026). It expects businesses that deploy agents to comply with consumer law and to monitor the agents for errors and bias. Meeting that expectation requires knowing what an agent's price preference is and where it came from.

We trace the origin of a price preference by following it through training. By origin we mean where in training the preference appears and how later stages change it. This needs a model whose training

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.

data and checkpoints, the copies saved during training, are public; OLMo 2 7B provides both (Team OLMo et al., 2025). At each checkpoint we elicit preferences with conjoint-style choice tasks.

We design a set of choice tasks around hotel booking, for three reasons. Corporate travel booking is well suited to an agent: it is repetitive and it runs at scale. The user's constraints, such as a city, dates and a budget ceiling, usually leave many options, so the final choice falls to the agent. Price and quality both matter in a hotel. And the setting resembles the e-commerce and booking testbeds already used to study how agents choose (Allouah et al., 2026; Cherep et al., 2026; Reusens et al., 2026). Asked to book a night in Manhattan, an agent must decide whether a 5-star suite with a spa is worth about $1,400 more than a well-reviewed 4-star room. In each task the model sees two such hotels that differ in price and other attributes, including star rating, bed size, free cancellation, breakfast, review score and review count, and must pick one.

We follow three outcomes separately across checkpoints. Representation is what numerical information can be read from the model's internal activations, and on what scale. Comparison is whether the model's answer scores favor the lower of two prices. Weighting is how much a displayed price moves a choice when the other attributes also vary. The price weight is the estimated strength of the preference for a lower log price in a discrete-choice model fitted to these tasks, with these six attributes as controls. Following the three separately shows how a model acquires its price preference, stage by stage. Weighting must be read against decisiveness, which is how firmly the model commits to one option when the two differ only in non-price attributes. We measure it by showing the same pairs with the price attribute removed and averaging the strength of the model's preference. A model that is barely swayed by stars or reviews has low decisiveness. The ratio of price weight to decisiveness is the relative price weight: how strongly price affects the choice compared with the model's responses to the non-price attributes on these tasks. Two further studies ask what in the training data could have produced this weighting: one audits the pretraining text for prices in contexts that settle a choice, and one audits the preference labels the model learned from for a preference for cheaper options. Together the five studies form a template for other models whose checkpoints and training data are available, and we release the code that runs them.

Our central finding is that accuracy in comparing prices does not reveal how strongly price influences a choice. At the first checkpoint we test, about 10% of the way through pretraining, the model already carries magnitude information fine enough to tell numbers of equal digit count apart, such as 200 and 800. Yet on questions asking which of two prices is lower, the model's scores initially favor the higher one more often than not. Comparison accuracy and price weighting both improve during pretraining, but high accuracy does not fix the strength of the preference. Across the 18 pretraining checkpoints with at least 97% accuracy, relative price weight ranges from 0.09 to 0.87. Once accuracy reaches 98%, a comparison measure that does not saturate and the price weight change by different factors at every step between neighboring checkpoints. The price weight favors the cheaper hotel at every checkpoint that compares prices accurately, so a preference for the cheaper hotel is present before any preference labels. In a sample of the web text the model was trained on, an LLM judge placed an estimated third of genuine detected prices in comparisons that help settle a choice, a larger share than for distances, populations or rating counts. Text of that kind is a plausible source, although the judge did not record which option each comparison favors.

Across the first 5B tokens of annealing, decisiveness more than doubles while the price weight shows no detectable change, so the relative price weight halves. In the plain prompt, supervised fine-tuning (SFT) repeats this on a larger scale. Decisiveness rises more than fourfold while the price weight shows no detectable change, so the relative price weight falls from about 0.40 to 0.12. Direct preference optimization (DPO) then roughly doubles relative price weight in the plain prompt, to about 0.26, partly reversing the decline. In the chat template the increase from SFT to DPO is much smaller. A targeted audit of price-related training pairs, with limited power, finds no detectable advantage for cheaper-favoring answers. At all four released checkpoints the response to price is closer to logarithmic than linear: under the logarithmic fit, a $50 cut counts about 4.7 times as much on a $150 room as on a $600 one. Post-training thus mainly changes the strength of a preference present before any preference labels. A business that must monitor its agent for bias therefore needs the training trajectory and data, not only the released model or its labels, to learn where the agent's price preference came from. Our contributions are as follows.

- **A measure of how much price counts.** Relative price weight divides the price weight by decisiveness on the same tasks without prices. It is needed because every SFT and DPO

step we measured, in five models, raised decisiveness, so raw weights mix preference with sharpness.

- **Comparison accuracy does not reveal price weighting.** Neither accuracy nor a comparison measure that does not saturate tracks the price weight across checkpoints, so a test of a capability does not certify a preference.
- **Auditable preferences form before preference labels.** In the plain prompt, the price weight favors the cheaper option wherever prices are compared accurately, including in three base models. The size of the later DPO step depends on the model, the prompt format and the training run (Appendix D.3).
- **What training adds to the number scale.** Untrained networks already show logarithmic spacing across the full range of values; training adds magnitude information within equal digit counts by the first checkpoint probed.
- **A reusable audit template.** The five studies can be rerun on other models with public checkpoints and training data; we rerun the weighting study on five other models, and its shuffled-price design is robust to the choice of controls, unlike a regression on listed prices.

2 SETUP

Table 1 maps the five studies to their methods and the access each needs.

We study one model lineage, OLMo 2 7B, whose training data and checkpoints are public (Team OLMo et al., 2025; Lambert et al., 2025). Pretraining has two stages. *Stage 1* uses 3.896T tokens of mostly web text; we test 15 checkpoints, starting at 399B. *Annealing* adds 50B tokens on a curated mixture. The developers ran three annealing runs with different data orders and averaged their final weights to produce *Base*. We follow intermediate checkpoints from one run to trace changes during annealing.

Post-training then adapts Base to answer users. Supervised fine-tuning (SFT) trains it to imitate example answers. Direct preference optimization (DPO) trains the SFT checkpoint to favor answers rated better in paired judgments. Instruct adds reinforcement learning to DPO. Base, SFT, DPO and Instruct thus name four released checkpoints along the same lineage; we call them the four *released stages*. We also score two *untrained networks* with the same architecture and random weights as references for what training adds.

Every checkpoint answers the same hotel booking tasks. Figure 1 shows one of the 600 tasks used to estimate price weight, with all displayed attributes. Every checkpoint is scored in the *plain prompt*: the task text followed by "Answer:". This provides a common format across training. The three post-trained stages are also scored in their *chat template*, which wraps the task as a user message and scores the start of the assistant's reply. This is the conversation format they were trained to use.

At the answer position we read the probabilities assigned to the letters A and B. Each task is scored twice, with the hotels exchanged, to account for a possible preference for one answer letter (Zheng et al., 2024). Let q_i^{AB} be task i with hotel 1 shown as A, and q_i^{BA} the same task with hotel 1 shown as B. We first compute the log-probability difference $\ell_i^o = \log[P(\mathrm{A} \mid q_i^o)/P(\mathrm{B} \mid q_i^o)]$ for each order $o \in \{AB, BA\}$, then average in hotel 1's direction: $d_i = (\ell_i^{AB} - \ell_i^{BA})/2$. The order-averaged choice log-odds d_i, which we call the *choice score*, is positive when the model favors hotel 1. A constant bias toward one position cancels in this average. We also record *slot-lock*, the share of tasks in which the same answer position wins in both orders.

In a binary logit model, choice log-odds equal the difference in the values assigned to the options from their attributes (McFadden, 1974). We use this relation to estimate the *price weight* w by least squares across the 600 tasks, fitting $d_i = -w \log(p_{i1}/p_{i2}) + \boldsymbol{\gamma}^\top \Delta \mathbf{x}_i + \epsilon_i$. Here p_{i1} and p_{i2} are the displayed hotel prices, and ϵ_i is the residual. The vector $\Delta \mathbf{x}_i$, with weights $\boldsymbol{\gamma}$, contains hotel 1 minus hotel 2 differences in six controls: star rating, bed-size rank (an ordered code for bed size), free cancellation, breakfast, review score and review count in hundreds. The minus sign makes a larger w a stronger preference for the cheaper hotel. To reduce the association between price and quality, displayed price pairs are shuffled among tasks of similar price level before the model is scored. The hotel descriptions remain fixed, including names, neighborhoods, amenities and room

Table 1: **The five studies of the audit template: questions, methods and data requirements.**

Study	Question	Method (access needed)
Representation	What numerical information can be read from hidden states, and on what scale?	A prompt showing two numbers, and a direction in the hidden state at the answer position along which the numbers are ordered (hidden states of checkpoints)
Comparison	Do the model's answer scores favor the lower of two prices?	303 two-price questions, scored in both orders (answer log-probabilities of checkpoints)
Weighting	How much does a displayed price move a choice when other attributes also vary?	600 hotel choices with displayed prices shuffled among tasks of similar price level (answer log-probabilities of released or intermediate checkpoints)
Corpus exposure	In what contexts does pretraining text present prices?	Sampled pretraining documents; one LLM judge marks price mentions and six other quantities against a rubric (pretraining text)
Preference labels	Do the labels DPO learned from carry a cheaper-favoring stance?	1,600 pairs audited from the 378,341-pair public preference set, in both orders (post-training data)

details beyond bed size that the six controls do not capture. We also refit with 71 controls, adding amenities and neighborhood indicators, to check sensitivity to omitted attributes.[1]

A larger price weight can reflect either a stronger response to price or sharper choices overall. In a logit model, a common scale factor multiplies every attribute coefficient (Swait & Louviere, 1993). We therefore measure *decisiveness* C on the same 600 tasks with both hotels' prices removed, as $C = (1/600)\sum_{i=1}^{600} |d_i^0|$, and define the relative price weight as $r = w/C$. The score d_i^0 is computed in the same way as d_i on the task without prices. Thus C measures the average strength of choices based on non-price attributes, and a larger *relative price weight* r means price matters more relative to them. If all choice log-odds double, both w and C double and r stays fixed; if non-price preferences strengthen while w stays fixed, r falls. We report w, C and r with 95% intervals from resampling the paired tasks, the *task bootstrap*. These intervals reflect task sampling, not variation between neighboring checkpoints or training runs. Neighboring checkpoints differ several times as much as task sampling predicts, so within a training phase we draw conclusions from patterns, not from single steps (Appendix A.4).

[1] In the plain prompt, from 2.937T through Instruct the price weight changes by less than 10% with 71 controls, and by at most 1.2% at the released checkpoints. A regression on the original listed prices depends strongly on the controls: at Base its price weight is 0.000 with six controls and 0.038 with 71 (Appendix A.2).

Option A

Hotel: New Yorker by Lotte Hotels
Star rating: 4 stars
Neighborhood: Midtown
Guest review score: 9.0/10
(14,872 reviews)
Room type: Deluxe King (1 king bed)
Free cancellation: No
Breakfast included: No
Key amenities: Free WiFi; Fitness center; Restaurant; Bar
Price per night: $125

Option B

Hotel: Conrad New York Midtown
Star rating: 5 stars
Neighborhood: Midtown
Guest review score: 8.5/10
(1,568 reviews)
Room type: Deluxe Suite (1 king bed)
Free cancellation: No
Breakfast included: No
Key amenities: Free WiFi; Fitness center; Restaurant; Bar; Spa
Price per night: $1,527

Figure 1: **A hotel choice task.** The plain prompt begins, "You are booking a hotel room in New York City for a one-night stay. You must choose between the following two options." It then lists the two options sequentially, shown here as panels, and asks "Which option do you choose? Reply with only the letter A or B." The cue "Answer:" marks the reply position. The prices are shuffled in from another task of similar price level; both option orders are scored.

3 PRICE RESPONSE AND REPRESENTATION

At the released checkpoints, the price response and the internal number scale are both closer to logarithmic than linear.

At every released checkpoint, the choice score falls as the displayed price rises in the plain prompt (Figure 2). Under the logarithmic term, a $50 cut from $150 moves the score about 4.7 times as much as a $50 cut from $600. People also give a fixed saving more weight when the price is lower (Thaler, 1980; Tversky & Kahneman, 1981). The logarithmic term predicts held-out scores better in all 20 cross-validation repetitions at each released checkpoint. It also wins in the chat template for the three post-trained models. A flexible five-bin price function fits better still after post-training, so the logarithmic form is an approximation (Appendix B.1).

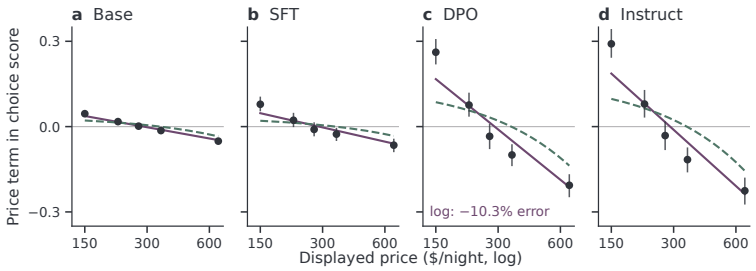


Figure 2: **Price response at the four released stages.** Curves show the centered price contribution to plain-prompt choice scores, with six attribute controls. Dots are price-quintile estimates with 95% task-bootstrap intervals. Solid and dashed lines are logarithmic and linear fits, respectively. The logarithmic term reduces held-out mean squared error by 3.6%–10.3% (Table A3).

Internal number scale. This response motivates asking whether the model's internal number scale is also logarithmic. We measure it with comparison prompts that have no hotel attributes. For example, a price prompt presents "Option A: Price: $151" and "Option B: Price: $603", then asks "Which offer is cheaper? Reply with only the letter A or B." We pair 40 values from 12 to 96,001 in both orders. Other versions present bare numbers, distances or populations (Appendix B.2).

We examine internal activations using a linear *readout*, a standard probing method (Belinkov, 2022). The readout projects the hidden state at the last prompt token, immediately before the model predicts an answer, onto a fitted direction. One method fits the direction to the largest pattern of activation change when the two numbers exchange positions. The other contrasts average states for pairs where the first number is much larger or much smaller than the second, the difference-of-means method (Marks & Tegmark, 2024).

To measure the scale, we ask whether the numerical difference or the log ratio better predicts the readout value. We fit these relations on some pairs and test them on held-out pairs made from the same values. *Held-out* R^2 is negative when the fit predicts worse than the mean baseline. This tests numerical structure in the activations, not whether the model uses it to answer correctly.

The logarithmic relation fits better in all 176 comparisons. These are $11 \times 4 \times 2 \times 2$ combinations: eleven checkpoints from 3.608T through Instruct, four numeral families, two layers (14 and 19), and the two readout methods. They share data and are not independent replications. Median held-out R^2 is 0.84 for the logarithmic relation and 0.39 for the linear one. A further fit uses a power transformation that includes the logarithm at exponent 0 and the linear scale at 1 (Box & Cox, 1964). Its exponents range from -0.04 to 0.18, near the logarithmic end (Appendix B.2). With the difference-of-means readout at layer 19, the price exponent is 0.04 at all four released checkpoints, showing stability across post-training at that readout site.

What training adds. With the layer-19 difference-of-means readout, the untrained networks' full-range logarithmic fits reach R^2 values of 0.69 and 0.71 on the weaker of the price and bare-number prompts. Their power exponents are also near zero (-0.02 to 0; Appendix B.2). Random weights can preserve structure in how inputs are written, which a fitted readout can recover. Decimal notation supplies a coarse logarithmic cue: 100, 1,000 and 10,000 each add one digit for a tenfold increase

in value. The logarithmic spacing over the full range is therefore present before training, and a high full-range fit alone does not show that training added numerical knowledge (Belinkov, 2022).

To test what training adds beyond this coarse cue, we restrict the comparison to pairs with the same digit count. Digit count can distinguish 200 from 2,000 but cannot distinguish 200 from 800. This *within-digit* test keeps that readout direction fixed and refits the relation between its value and the log ratio using only same-length pairs. The untrained values are -0.07 and -0.12; this readout fails the mean baseline. At 399B, the earliest checkpoint probed, the trained model reaches 0.39. Figure 3a follows this within-digit measure across training. It is positive at every trained checkpoint measured, though it does not improve steadily through stage 1. Training therefore adds within-digit magnitude information by the first checkpoint probed, long before any preference labels, while the full-range logarithmic spacing is already present before training.

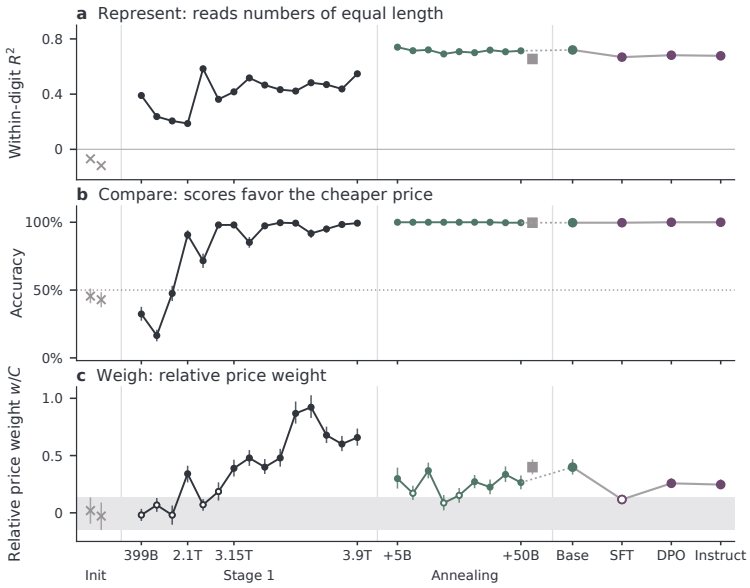


Figure 3: **Representation, comparison and weighting across training.** (a) Within-digit held-out R^2 of the layer-19 difference-of-means readout, the weaker of the price and bare-number values. (b) Share of price-comparison questions whose scores favor the cheaper price. (c) Relative price weight w/C. Panels b–c use the plain prompt and show 95% task-bootstrap intervals; a shows point estimates. Crosses mark untrained networks; squares mark a second annealing run. The dotted line in b marks chance. The gray band in c spans the untrained intervals; hollow circles have intervals overlapping it. Spacing follows checkpoint order within each phase, not token counts.

4 A PRICE PREFERENCE DEVELOPS DURING PRETRAINING

Pretraining develops both accurate price comparison and a preference for cheaper hotels, before any preference labels are used. A price preference might simply follow from learning to compare prices. The two do improve together over stage 1, but comparison accuracy alone does not reveal how strongly price influences a hotel choice (Figure 3b–c).

Comparison accuracy is the share of 303 two-price questions whose order-averaged choice score favors the cheaper price. Accuracy starts below chance, at 32.3% at 399B, despite the numerical information already present in the model's activations. It reaches 98.0% by 2.937T, fluctuates at later checkpoints, and ends stage 1 at 99.3%. Accurate comparison is therefore present well before the end of pretraining, although performance does not improve steadily.

A preference for the cheaper hotel forms during stage 1. The raw price weight rises with training (Kendall's $\tau = 0.73$ across the 15 stage-1 checkpoints; Appendix C.1) while decisiveness does not rise ($\tau = -0.24$), and the price weight favors the cheaper hotel, with its interval above zero, at every checkpoint that compares prices accurately. These stage-1 scores are conditional on a letter answer,

because A and B hold under 63% of the answer probability (Table A4), and first-token letter scores can differ from generated answers (Wang et al., 2024). Because decisiveness is lower at stage 1 than in the untrained networks, relative price weight clears its untrained band at every checkpoint from 3.146T to 3.896T, whereas the raw weight clears its own at only three (Appendix C.2).

High comparison accuracy can coexist with substantially different price weights. Across the 18 pretraining checkpoints with at least 97% accuracy, relative price weight ranges from 0.09 to 0.87 (Figure 3c). Because accuracy saturates, we also measure the *comparison slope*, the change in the order-averaged choice score per unit of log price ratio; it does not saturate and can be compared with the raw price weight without normalization. At all 18 steps between neighboring checkpoints from 2.937T to the end of annealing, the two change by different factors, sometimes in opposite directions, and across the 26 trained pretraining checkpoints they are only weakly rank-correlated (Spearman 0.37; Appendix C.3). An audit therefore needs to measure the price–quality trade-off directly, even when the model compares prices accurately.

Annealing changes this trade-off mainly through the non-price attributes. Across its first 5B tokens, the estimated price weight is nearly unchanged, while decisiveness on tasks without prices more than doubles, from 0.053 to 0.123. Relative price weight consequently falls from 0.66 to 0.30, a fall of 0.36 [0.25, 0.46]. In this run the comparison slope rises 3.4-fold [3.3, 3.6] over the same tokens; the second annealing run shows no such rise (Appendix C.3). The same pattern holds across all nine annealing checkpoints. Their median decisiveness is about twice that of the nine stage-1 checkpoints from 3.146T, while their median price weight is about the same. The saved no-price fits also give larger coefficients to star rating and included breakfast, consistent with stronger responses to hotel quality cues (Appendix C.4).

After 50B annealing tokens, relative price weight is 0.26 in the run we follow and 0.40 in a second run on the same data in a different order. Both are below the 0.66 at which annealing began, but the final trade-off varies between runs. The released Base model, which averages the weights of three annealing runs, has a relative price weight of 0.40.

We also examined whether pretraining text presents price as a reason to choose. In an exploratory audit of stage-1 web text, an LLM judge confirmed 87 sampled price mentions. After adjustment for sampling rates, an estimated 31.8% [19.0, 46.7] of them occurred in comparisons that help settle a choice. This share is higher than for distances, populations or rating counts, but not clearly higher than for car model years (Appendix E.1). Because the price preference predates any preference labels, a search for its source needs the pretraining data, not only the released model.

5 Post-training changes an existing preference

Post-training changes a price preference already present in Base. Table 2 gives w, C, r and slot-lock for each released stage and format.

Table 2: **Price weight and decisiveness across released checkpoints.** All columns use the 600 shuffled-price tasks and their no-price versions. Brackets are 95% task-bootstrap intervals. Base has no chat template; its placebo score is in Appendix A.3.

Stage	w [95%]	C [95%]	$r = w/C$ [95%]	Slot-lock
Plain prompt				
Base	0.057 [0.048, 0.067]	0.143 [0.136, 0.150]	0.399 [0.334, 0.469]	0.622
SFT	0.072 [0.049, 0.096]	0.620 [0.589, 0.652]	0.116 [0.079, 0.154]	0.997
DPO	0.255 [0.218, 0.297]	0.991 [0.942, 1.040]	0.258 [0.217, 0.301]	0.945
Instruct	0.285 [0.242, 0.333]	1.157 [1.099, 1.215]	0.247 [0.207, 0.291]	0.908
Chat template				
SFT	0.218 [0.181, 0.256]	0.805 [0.764, 0.850]	0.270 [0.223, 0.322]	0.968
DPO	0.416 [0.358, 0.480]	1.443 [1.373, 1.518]	0.289 [0.245, 0.336]	0.452
Instruct	0.335 [0.285, 0.390]	1.200 [1.137, 1.267]	0.279 [0.235, 0.328]	0.477

Interpreting choice scores. High slot-lock means the same letter, A or B, has the higher probability even when the two hotels exchange positions. In the plain prompt this happens on over 90% of tasks

after post-training. At every released stage in the plain prompt, the higher-scored letter also holds more probability than all tokens other than A and B combined, on every hotel prompt, so greedy decoding would begin with that letter. Price can still change the probability margin between A and B without reversing their ranking, and the order-averaged price weight measures these changes. The estimates therefore describe sensitivity in the scores, not how often sampled recommendations would change. Format alone does not explain the lock, because SFT is also locked on 96.8% of chat-template tasks. After DPO this falls to 45.2%, although much slot-lock remains (Appendix D.2).

Changes in relative price weight. SFT raises decisiveness much more than price weight. In the plain prompt, decisiveness rises 4.3-fold from Base to SFT, while the price weight shows no detectable change (1.26-fold, paired interval [0.95, 1.57]). Relative price weight therefore falls from 0.399 to 0.116. SFT does not lower the price weight; it makes the other attributes matter more (Appendix D.1, Appendix D.4). An audit of the price weight alone would miss this change.

DPO trains on recorded judgments between answers (Rafailov et al., 2023). In the plain prompt it partly reverses the SFT decline, mainly by raising the price weight. The price weight rises 3.54-fold and decisiveness 1.60-fold, so relative price weight rises 2.21-fold [1.90, 2.83], and it also rises under two other measures of decisiveness (Appendix D.1). In the chat template the increase is 1.07-fold [1.02, 1.13], or 1.55-fold [1.41, 1.81] when the system message "You are a helpful assistant." is added (Appendix D.3). Instruct is similar to DPO. After DPO, the plain-prompt relative price weight (0.26) remains below its values at the end of stage 1 (0.66) and at Base (0.40). This comparison depends on how decisiveness is measured. On 400 pairs of identical products that differ only in a listing number, the DPO value is above both levels. On the original hotel tasks, it is below the stage-1 level, and its comparison with Base is inconclusive (Appendix D.1).

Which patterns generalize. We reran the weighting study on the released checkpoints of five other models and on four DPO runs retrained from the OLMo 2 7B SFT checkpoint on 64,000-pair samples of its preference data (Appendix D.3, Appendix E.3). Every SFT and DPO step we measured raised decisiveness, and DPO raised the raw price weight in all but one case, so raw weights mix preference with sharpness. Dividing by decisiveness narrows differences between training objectives. Seven preference objectives trained from one Llama 3 8B SFT checkpoint differ 5.9-fold in price weight but 1.5-fold in relative price weight. In the four other models where the DPO step can be measured, it is smaller than the plain-prompt increase in OLMo 2 7B, and its size depends on the prompt format, a system message and the training run. The four retrained runs raised the plain-prompt relative price weight 1.45- to 2.84-fold, including both runs trained without price-related pairs, and in each run the chat-template change was smaller.

We also audited whether DPO labels favor cheaper answers. In 1,600 training pairs sampled to emphasize price comparisons, chosen answers show no detectable excess of cheaper-favoring language over rejected answers: a difference of 0.16 percentage points [-0.88, 1.19]. The audit has limited power and concerns these selected pairs (Appendix E.2). In the plain prompt, the price response stays closer to logarithmic than linear at every released stage (Table A3), and the layer-19 difference-of-means exponent of the internal price readout stays at 0.04 (Section 3).

6 Related work

Economic behavior of language-model agents. Released models show measurable economic rationality and preferences (Chen et al., 2023; Raman et al., 2024; Ross et al., 2024; Goli & Singh, 2024; Brand et al., 2026; Mazeika et al., 2025), trade price against quality as shopping and booking agents (Allouah et al., 2026; Cherep et al., 2026; Reusens et al., 2026), and assume that people are more rational than they are (Liu et al., 2025). Language-model pricing agents, seller-inserted text and agent-assisted search can affect markets (Fish et al., 2026; Kumar & Lakkaraju, 2024; Lucier et al., 2026). The empirical studies among these measure a released model after training; none follows the training that set the weight.

Numeric representations. Numerical magnitude can be read from embeddings and hidden states (Wallace et al., 2019; Zhu et al., 2025a), as can real-valued quantities such as locations and dates (Heinzerling & Inui, 2024; Gurnee & Tegmark, 2024), but studies disagree about the code. Zhu et al. (2025a) decode log values linearly but inexactly, Levy & Geva (2025) find a digit-wise base-10 code, Engels et al. (2025) find circular features for periodic quantities, and tokenization shapes

arithmetic (Singh & Strouse, 2024). Human-like ratio effects (Shah et al., 2023) and a roughly logarithmic number line, as proposed for people (Dehaene, 2003), have been reported in released LLMs (AlQuabeh et al., 2025), and a small language model trained only on arithmetic predicts human risky and intertemporal choice (Zhu et al., 2025b). What a readout shows depends on its baseline (Hewitt & Liang, 2019; Belinkov, 2022), so Section 3 adds untrained-network and within-digit controls across training.

Interpretability across training stages. Public checkpoint suites make training observable (Biderman et al., 2023; Team OLMo et al., 2025), and studies of training checkpoints report when a capability appears, which mechanism carries it, and how it depends on pretraining data (Chen et al., 2024; Tigges et al., 2024; Merullo et al., 2025; Kim et al., 2025). Fine-tuning reportedly acts on mechanisms and knowledge the base model already has (Prakash et al., 2024; Lee et al., 2024; Lin et al., 2024), and DPO can concentrate probability on outputs that are already likely (Ren & Sutherland, 2025), consistent with the rises in decisiveness we measure. Cognitive biases and reward-model value biases trace back to pretraining (Itzhak et al., 2025; Christian et al., 2026), while post-training can shift expressed values (Bhatia et al., 2026). We follow one economic preference through training and read it against decisiveness rather than a task score.

7 Discussion and conclusion

Economic preferences in language models need not begin with preference optimization. In this lineage, a preference for cheaper hotels develops during pretraining, alongside improving comparison accuracy, and the choice scores favor the cheaper option wherever the model compares prices accurately. Neither comparison accuracy nor the comparison slope specifies the strength of that preference. Annealing and, in the plain prompt, SFT lower the relative price weight by strengthening choices based on non-price attributes, and DPO partly restores it. SFT and DPO raised decisiveness in every model and run we measured, while the size of the DPO step in relative price weight varies (Appendix D.3). The price response of OLMo 2 7B and the scale recovered from its activations are both closer to logarithmic than linear. Because untrained networks share the full-range logarithmic spacing, this correspondence is consistent with, but does not show, an internal number scale shaping how price enters the choice score.

Three consequences follow. First, under a logarithmic price response, a seller of an expensive room would need a larger dollar discount than a seller of a cheap room to move the agent's scores equally. Second, audits must name format and version, because the size of a post-training change depends on the prompt format, the model and even the training run. They should report the price weight together with decisiveness on the same choices with the price attribute removed, because SFT and DPO raise decisiveness, so raw attribute weights mix preference with sharpness. They should also report slot-lock. This part needs only answer log-probabilities, so it can run on a deployed model that exposes them but not its checkpoints. Third, because the price weight precedes the preference labels, pretraining data belong in the audit beside them.

Limitations and future research. First, the training trajectory and data audits suggest how the preference develops, but do not isolate its causes. Retraining at this scale is costly, and our small DPO retraining experiment raised the price weight with and without price-related pairs but has too few seeds to test their role (Appendix E.3). Future work could continue training from a common checkpoint with and without price-related text or preference labels, across seeds, and could intervene on the internal number scale. Second, we trace the full training sequence of one 7B model and only the released checkpoints of five others (Appendix D.3); the study template can test which patterns recur in other sizes, architectures and training recipes. Third, the estimates depend on the hotel tasks, displayed prices and prompt formats. They use answer probabilities rather than generated recommendations, so future work should vary products, prices and wording and evaluate recommendations where agents are deployed.

Agents are beginning to book and pay for users, and the weight they give price can affect what users pay. The five studies, whose code we release, measure that weight and trace where in training it formed.

ETHICS STATEMENT

This research uses publicly released models and training data and involves no human participants. Measuring an agent's price preferences can help auditors assess whether it serves users' interests. The same methods could help sellers personalize prices or offers to those preferences, potentially increasing what users pay without improving what they receive. This raises concerns about fairness and transparency, especially when users cannot see or control how an agent trades price against other attributes on their behalf.

REPRODUCIBILITY STATEMENT

Appendix A documents the tasks, checkpoints and estimation procedures. The study-specific appendices provide additional methods and robustness checks. We will release the audit template on GitHub, with reproduction instructions. It contains the 4,006 choice and comparison items and the number-pair prompts used for the readouts; the code that scores a checkpoint and estimates each study's quantities, including w, C, r, slot-lock and their task-bootstrap intervals; the judge rubrics for the two data audits; a configuration file that lists another model's checkpoints, with OLMo 2 7B as the worked example; and self-tests that run without a model.

AI USE STATEMENT

AI tools, including Claude Code, OpenAI Codex and Cursor Composer 2.5, assisted with study design, implementation, data annotation and analysis, interpretation, literature review and manuscript preparation under author supervision. The authors verified the analyses, numbers and references and take full responsibility for the content.

REFERENCES

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. doi: 10.1109/TAC.1974.1100705. URL https://doi.org/10.1109/TAC.1974.1100705.

Amine Allouah, Omar Besbes, Josué D. Figueroa, Yash Kanoria, and Akshit Kumar. What Is Your AI Agent Buying? Evaluation, Biases, Model Dependence, & Emerging Implications of Agentic E-Commerce. In *Proceedings of the ACM Web Conference 2026*, pp. 8697–8700, 2026. doi: 10.1145/3774904.3792943. URL https://doi.org/10.1145/3774904.3792943. Full version: arXiv:2508.02630.

Hilal AlQuabeh, Velibor Bojkovic, Tatsuya Hiraoka, Ahmed Oumar El-Shangiti, Munachiso Nwadike, and Kentaro Inui. Number Representations in LLMs: A Computational Parallel to Human Perception, 2025. URL https://arxiv.org/abs/2502.16147. arXiv preprint arXiv:2502.16147.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Mehar Bhatia, Shravan Nayak, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, Vered Shwartz, and Siva Reddy. Value Drifts: Tracing Value Alignment During LLM Post-Training. *Transactions of the Association for Computational Linguistics*, 14:2033–2056, 2026. doi: 10.1162/tacl.a.790. URL https://aclanthology.org/2026.tacl-1.92/.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory 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. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

George E. P. Box and David R. Cox. An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964. doi: 10.1111/j.2517-6161.1964.tb00553.x. URL https://doi.org/10.1111/j.2517-6161.1964.tb00553.x.

James Brand, Ayelet Israeli, and Donald Ngwe. Using LLMs for Market Research. Working Paper 23-062, Harvard Business School, 2026. URL https://doi.org/10.2139/ssrn.4395751. Forthcoming in *Quantitative Marketing and Economics*. Previously circulated as "Using GPT for Market Research"; extended abstract in EC '24.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. Sudden Drops in the Loss: Syntax Acquisition, Phase Transitions, and Simplicity Bias in MLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MO5PiKHELW.

Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of GPT. *Proceedings of the National Academy of Sciences*, 120(51):e2316205120, 2023. doi: 10.1073/pnas.2316205120. URL https://doi.org/10.1073/pnas.2316205120.

Manuel Cherep, Chengtian Ma, Abigail Xu, Maya Shaked, Pattie Maes, and Nikhil Singh. A Framework for Studying AI Agent Behavior: Evidence from Consumer Choice Experiments. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xAPoscV2Bw.

Brian Christian, Jessica A. F. Thompson, Elle Michelle Yang, Vincent Adam, Hannah Rose Kirk, Christopher Summerfield, and Tsvetomira Dumbalska. Reward Models Inherit Value Biases from Pretraining. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dT399j1Azv.

Circle Internet Group. Circle Launches AI Infrastructure to Power the Agentic Economy, 2026. URL https://www.circle.com/pressroom/circle-launches-ai-infrastructure-to-power-the-agentic-economy. Circle pressroom, 11 May 2026, announcing Circle Agent Stack, including Agent Wallets that let an agent hold, send and manage funds within predefined guardrails. Accessed 20 September 2026.

Competition and Markets Authority. Agentic AI and consumers, 2026. URL https://www.gov.uk/government/publications/agentic-ai-and-consumers. UK Competition and Markets Authority, research and analysis, 9 March 2026, published with the companion guidance "Using AI agents: complying with consumer law". Accessed 17 September 2026.

Stanislas Dehaene. The neural basis of the Weber–Fechner law: a logarithmic mental number line. *Trends in Cognitive Sciences*, 7(4):145–147, 2003. doi: 10.1016/S1364-6613(03)00055-X. URL https://doi.org/10.1016/S1364-6613(03)00055-X.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not All Language Model Features Are One-Dimensionally Linear. In *The Thirteenth International Conference on Learning Representations*, pp. 84591–84622, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic Collusion by Large Language Models. In *Proceedings of the 27th ACM Conference on Economics and Computation*, 2026. URL https://arxiv.org/abs/2404.00806. Full version: arXiv:2404.00806.

Ali Goli and Amandeep Singh. Frontiers: Can Large Language Models Capture Human Preferences? *Marketing Science*, 43(4):709–722, 2024. doi: 10.1287/mksc.2023.0306. URL https://pubsonline.informs.org/doi/10.1287/mksc.2023.0306.

Aaron Grattafiori et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783. arXiv preprint arXiv:2407.21783.

Paul E. Green and Vithala R. Rao. Conjoint Measurement for Quantifying Judgmental Data. *Journal of Marketing Research*, 8(3):355–363, 1971. doi: 10.1177/002224377100800312. URL https://doi.org/10.1177/002224377100800312.

George Gui and Olivier Toubia. The Challenge of Using LLMs to Simulate Human Behavior: A Causal Inference Perspective, 2023. URL https://arxiv.org/abs/2312.15524. arXiv preprint arXiv:2312.15524.

Wes Gurnee and Max Tegmark. Language Models Represent Space and Time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jE8xbmvFin.

Benjamin Heinzerling and Kentaro Inui. Monotonic Representation of Numeric Attributes in Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 175–195. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-short.18. URL https://aclanthology.org/2024.acl-short.18/.

John Hewitt and Percy Liang. 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)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Itay Itzhak, Yonatan Belinkov, and Gabriel Stanovsky. Planted in Pretraining, Swayed by Finetuning: A Case Study on the Origins of Cognitive Biases in LLMs. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=KQhUEoPmJy.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825. arXiv preprint arXiv:2310.06825.

Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. Knowledge Entropy Decay during Language Model Pretraining Hinders New Knowledge Acquisition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eHehzSDUFp.

Pavel Kireyev. PriceBench: A Diagnostic Benchmark for Price, Quality, and Brand Preferences in LLM Booking Agents. SSRN Working Paper 7452458, 2026. URL https://doi.org/10.2139/ssrn.7452458.

Aounon Kumar and Himabindu Lakkaraju. Manipulating Large Language Models to Increase Product Visibility, 2024. URL https://arxiv.org/abs/2404.07981. arXiv preprint arXiv:2404.07981.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/pdf/9c0d921fcc7c90227f3e1bcba16bfea6f63ad480.pdf.

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A Mechanistic Understanding of Alignment Algorithms: A Case Study on DPO and Toxicity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 26361–26378. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24a.html.

Amit Arnold Levy and Mor Geva. Language Models Encode Numbers Using Digit Representations in Base 10. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 385–395. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-short.33. URL https://aclanthology.org/2025.naacl-short.33/.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, pp. 14200–14282, 2024. doi: 10.52202/079017-0455. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets_and_Benchmarks_Track.html.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The Unlocking Spell on Base LLMs: Rethinking Alignment via In-Context Learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6bcbb4a501dbad0eba1b660c1a55318c-Abstract-Conference.html.

Ryan Liu, Jiayi Geng, Joshua C. Peterson, Ilia Sucholutsky, and Thomas L. Griffiths. Large Language Models Assume People are More Rational than We Really are. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dAeET8gxqg.

Jordan J. Louviere and George Woodworth. Design and Analysis of Simulated Consumer Choice or Allocation Experiments: An Approach Based on Aggregate Data. *Journal of Marketing Research*, 20(4):350–367, 1983. doi: 10.1177/002224378302000403. URL https://doi.org/10.1177/002224378302000403.

Brendan Lucier, Nicole Immorlica, Markus Mobius, Aleksandrs Slivkins, Daniel G. Goldstein, Jake M. Hofman, Sonia Jaffe, and David M. Rothschild. Agentic Markets: Equilibrium Effects of Improving Consumer Search, 2026. URL https://arxiv.org/abs/2603.25893. arXiv preprint arXiv:2603.25893.

Samuel Marks and Max Tegmark. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Mastercard. Mastercard unveils Agent Pay, pioneering agentic payments technology to power commerce in the age of AI, 2025. URL https://www.mastercard.com/global/en/news-and-trends/press/2025/april/mastercard-unveils-agent-pay-pioneering-agentic-payments-technology-to-power-commerce-in-the-age-of-ai.html. Mastercard newsroom press release, 29 April 2025, introducing Mastercard Agent Pay and Agentic Tokens for agent-initiated transactions. Accessed 20 September 2026.

Mantas Mazeika, Xuwang Yin, Rishub Tamirisa, Jaehyuk Lim, Bruce W. Lee, Richard Ren, Long Phan, Norman Mu, Oliver Zhang, and Dan Hendrycks. Utility Engineering: Analyzing and Controlling Emergent Value Systems in AIs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4705. URL https://openreview.net/forum?id=x9vcgXmRD0.

Daniel McFadden. Conditional Logit Analysis of Qualitative Choice Behavior. In Paul Zarembka (ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press, New York, 1974. URL https://eml.berkeley.edu/reprints/mcfadden/zarembka.pdf.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124198–124235, 2024. doi: 10.52202/079017-3946. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e099c1c9699814af0be873a175361713-Abstract-Conference.html.

Jack Merullo, Noah A. Smith, Sarah Wiegreffe, and Yanai Elazar. On Linear Representations and Pretraining Data Frequency in Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EDoD3DgivF.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-Tuning Enhances Existing Mechanisms: A Case Study on Entity Tracking. In *The Twelfth International Conference on Learning Representations*, pp. 8057–8082, 2024. URL https://openreview.net/forum?id=8sKcAWOf2D.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Narun Krishnamurthi Raman, Taylor Lundy, Samuel Joseph Amouyal, Yoav Levine, Kevin Leyton-Brown, and Moshe Tennenholtz. STEER: Assessing the Economic Rationality of Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 42026–42047. PMLR, 2024. URL https://proceedings.mlr.press/v235/raman24b.html.

Yi Ren and Danica J. Sutherland. Learning Dynamics of LLM Finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

Manon Reusens, Sofie Goethals, Toon Calders, and David Martens. Would a Large Language Model Pay Extra for a View? Inferring Willingness to Pay from Subjective Choices. *Expert Systems with Applications*, 331:133279, 2026. doi: 10.1016/j.eswa.2026.133279. URL https://doi.org/10.1016/j.eswa.2026.133279.

Jillian Ross, Yoon Kim, and Andrew W. Lo. LLM economicus? Mapping the Behavioral Biases of LLMs via Utility Theory. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Rx3wC8sCTJ.

Raj Shah, Vijay Marupudi, Reba Koenen, Khushi Bhardwaj, and Sashank Varma. Numeric Magnitude Comparison Effects in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6147–6161. Association for Computational

Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.383. URL https://aclanthology.org/2023.findings-acl.383/.

Shopify. The agentic commerce platform: Shopify connects any merchant to every AI conversation, 2026. URL https://www.shopify.com/news/ai-commerce-at-scale. Shopify newsroom, 11 January 2026; announces the Universal Commerce Protocol, co-developed with Google, under which "agents can natively complete checkout on a customer's behalf". Accessed 20 September 2026.

Aaditya K. Singh and DJ Strouse. Tokenization counts: the impact of tokenization on arithmetic in frontier LLMs, 2024. URL https://arxiv.org/abs/2402.14903. arXiv preprint arXiv:2402.14903.

Stripe. Stripe powers Instant Checkout in ChatGPT and releases Agentic Commerce Protocol codeveloped with OpenAI, 2025. URL https://stripe.com/newsroom/news/stripe-openai-instant-checkout. Stripe newsroom press release, 29 September 2025. Accessed 17 September 2026.

Joffre Swait and Jordan Louviere. The Role of the Scale Parameter in the Estimation and Comparison of Multinomial Logit Models. *Journal of Marketing Research*, 30(3):305–314, 1993. doi: 10.1177/002224379303000303. URL https://doi.org/10.1177/002224379303000303.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961. arXiv preprint arXiv:2512.13961.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 Furious (COLM's Version). In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU. Extended technical report: arXiv:2501.00656.

Richard Thaler. Toward a positive theory of consumer choice. *Journal of Economic Behavior & Organization*, 1(1):39–60, 1980. doi: 10.1016/0167-2681(80)90051-7. URL https://doi.org/10.1016/0167-2681(80)90051-7.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. LLM Circuit Analyses Are Consistent Across Training and Scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 40699–40731, 2024. doi: 10.52202/079017-1287. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/47c7edadfee365b394b2a3bd416048da-Abstract-Conference.html.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct Distillation of LM Alignment. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aKkAwZB6JV.

Amos Tversky and Daniel Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 211(4481):453–458, 1981. doi: 10.1126/science.7455683. URL https://doi.org/10.1126/science.7455683.

Visa. Find and Buy with AI: Visa Unveils New Era of Commerce, 2025. URL https://usa.visa.com/about-visa/newsroom/press-releases.releaseId.21361.html. Visa newsroom press release, 30 April 2025, introducing Visa Intelligent Commerce. Accessed 20 September 2026.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. 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)*, pp. 5307–5315. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534/.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. "My Answer is C": First-Token Probabilities Do Not Match Text Answers in Instruction-Tuned Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7407–7416. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.441. URL https://aclanthology.org/2024.findings-acl.441/.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large Language Models Are Not Robust Multiple Choice Selectors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=shr9PXz7T0.

Fangwei Zhu, Damai Dai, and Zhifang Sui. Language Models Encode the Value of Numbers Linearly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 693–709, Abu Dhabi, UAE, 2025a. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.47/.

Jian-Qiao Zhu, Haijiang Yan, and Thomas L. Griffiths. Language Models Trained to do Arithmetic Predict Human Risky and Intertemporal Choice. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Tn8EQIFIMQ.

A TASKS, ESTIMATION AND UNCERTAINTY

This appendix documents what the studies share. Appendix A.1 lists the task blocks and the checkpoints each study scores. Appendix A.2 explains why the price weight is estimated on shuffled prices and how the shuffle works. Appendix A.3 specifies how the prompt formats are scored and which measurements count as valid. Appendix A.4 explains what the task-bootstrap intervals measure and why conclusions within a training phase rest on patterns across checkpoints.

A.1 TASKS AND CHECKPOINTS

The tasks draw on a fixed pool of 179 New York hotel profiles. Hotel names, stars, neighborhoods and amenities are fixed within a profile. Prices, room types, cancellation terms, breakfast, review scores and review counts vary within hotel-specific ranges calibrated to listings. Table A1 describes the task collection. The 1,800 original hotel tasks come from 450 hotel pairings, each drawn four times with re-drawn prices and other variable attributes. The 600 shuffled-price tasks are a subset of them, with their displayed prices reassigned; each has a matching version with prices removed. All tasks are presented in both option orders. The internal readouts use a separate number-pair grid (Appendix B.2), and the check on other numerical attributes uses 2,800 further tasks (Appendix D.4); Table A1 counts neither.

Table A1: **Choice-task collection.** Counts are before exchanging option order. The listing-number block uses identical products; the listing-number tasks of Appendix D.4 are a separate set.

Task block	Count	Use
Hotel pairs at listed prices	1,800	Source of the shuffled tasks; A/B-mass condition of the validity rule
Pairs with shuffled prices	600	Price weight w and slot-lock; at their listed prices, the listed-price check and original-task normalization
Matching pairs without prices	600	Decisiveness C
Identical products with different listing numbers	400	Listing-number normalization
Two-price comparisons	303	Comparison accuracy, comparison slope and the comparison condition of the validity rule
Matching bare-number comparisons	303	Comparison without price wording (scored, not reported)
Total	4,006	

The main trajectory covers 26 pretraining checkpoints: 15 stage-1 checkpoints from 399B to 3.896T tokens, nine checkpoints from one annealing run, which we call the tracked run, the endpoint of a second annealing run, and Base. SFT, DPO and Instruct add three released checkpoints. Two networks initialized with random weights provide the untrained reference. The internal-scale comparison uses eleven checkpoints (Appendix B.2), two of which (3.608T and 3.754T) lie outside the main trajectory, and the check on other numerical attributes uses six (Appendix D.4). Appendix D.3 lists the other models scored. The model report's total of 4.05T tokens counts all three annealing runs (Team OLMo et al., 2025). One training path, such as the tracked run, has 3.896T stage-1 tokens plus 50B annealing tokens.

A.2 WHY PRICES ARE SHUFFLED

Listed prices track hotel quality that the six controls do not capture, so a regression on listed prices depends on which controls it includes. We therefore estimate the price weight on displayed prices that are shuffled among tasks. Table A2 compares the two designs at the released checkpoints. On listed prices the price weight depends strongly on the controls. With 71 controls instead of six it rises from 0.000 to 0.038 at Base and from 0.007 to 0.130 at SFT, so a listed-price audit with 71 controls would report a 3.40-fold [2.41, 4.64] rise from Base to SFT, where the shuffled design shows no detectable change. A placebo test shows why. The no-price tasks hide the price, yet with six controls their choice scores favor the hotel with the higher listed price at every released stage, so the listed price carries quality information that the six controls miss. With 71 controls this

association is no longer detectable. The shuffled-price weight barely changes with the controls, so it does not depend on knowing which attributes carry quality.

Table A2: **Shuffled versus listed prices.** Price weight w at the released checkpoints in the plain prompt, estimated on the 600 tasks with shuffled prices (the paper's design) or with the original listed prices, with the six controls of Section 2 or with 71 controls. The placebo is the coefficient from regressing the no-price scores on the unseen listed price; a positive value favors the hotel with the higher listed price. *The 95% task-bootstrap interval excludes zero.

	Shuffled prices		Listed prices		No-price placebo	
Stage	6 controls	71 controls	6 controls	71 controls	6 controls	71 controls
Base	0.057*	0.056*	0.000	0.038*	0.024*	0.001
SFT	0.072*	0.072*	0.007	0.130*	0.174*	0.010
DPO	0.255*	0.257*	0.222*	0.449*	0.285*	0.030
Instruct	0.285*	0.282*	0.269*	0.546*	0.308*	0.002

Displayed prices are shuffled within ten price-level groups, defined by each task's mean log listed price. A task's two displayed prices move together to another task in the same group, so each task keeps a similar price level but receives a different price gap. The permutation is chosen to reduce the association between the listed-price gap and the displayed-price gap that remains after adjusting for the six controls. Other attributes keep their original values. The price weight therefore describes responses under this within-group shuffle, not under independent randomization of every attribute. Shuffling within groups leaves some association between products and price levels, which limits causal interpretation of the price weight and of the fitted price response in Figure 2. The prompt also does not say that prices were reassigned, so the model may read a price as a signal of unstated quality (Gui & Toubia, 2023).

The regression has no intercept, because exchanging the two hotels reverses the sign of the score and of every attribute difference. The six-control fit codes cancellation and breakfast as binary indicators, bed size by rank and review count in hundreds; stars and review score enter on their displayed scales. The 71-control fit adds 47 amenity indicators, an amenity count and 17 neighborhood indicators to these six controls; 67 of its 71 control columns are linearly independent.

The shuffled-price weight is also insensitive to the controls across training and across models. In the plain prompt, the 71-control fit changes it by at most 0.005 at every trained checkpoint. From 2.937T through Instruct the change lies between -9.2% and $+2.2\%$, and it is at most 1.2% at the four released checkpoints. Before 2.937T the relative changes range from -57% to $+261\%$, although the absolute change stays below 0.005. Wherever the six-control weight is at least 0.015, the change lies within this -9.2% to $+2.2\%$ range. Across the 27 valid measurements of the other models (Appendix A.3, Appendix D.3), the change is at most 2.1% in 26; the exception is Zephyr 7B SFT in the plain prompt, whose weight of 0.028 changes by -16.7%.

A.3 Prompt formats and valid measurements

The plain prompt ends with the "Answer:" cue shown in Figure 1, and its answer labels are the space-prefixed A and B tokens. The chat template places the task in a user message and scores the bare A and B tokens at the start of the assistant's response, without that cue. Two variants of the chat template, used only for OLMo 2 models, are described in Appendix D.3. Tokenization checks cover the complete prompt and label continuations. Local scoring loads FP32 weights; GPU runs may use TF32 matrix multiplication, as recorded in their run metadata.

We call one checkpoint scored in one format a measurement. A measurement is valid if the checkpoint puts at least half of its answer probability on A or B, averaged over the 1,800 original hotel tasks, and favors the cheaper price on at least 90% of the two-price comparisons. The first condition requires that the model answer with a letter, and the second that it compare prices correctly. Slot-lock is reported but does not exclude a measurement. The rule selects the measurements in Appendix D.3, including those of the released OLMo 2 7B checkpoints. We do not apply it to the OLMo 2 7B pretraining checkpoints. Most stage-1 checkpoints would fail it, so their scores are read conditional on a letter answer (Appendix C.1).

Base has no chat template, so we score it with the SFT template as a format placebo, which shows what scoring in a format the model was not trained on produces. Base puts 2% of its probability on A or B in this format, far below the validity threshold, and every task is slot-locked. The measurement is therefore not valid, and Table 2 has no chat-template row for Base. For reference, its price weight is -0.012 [-0.017, -0.006], its decisiveness 0.047 [0.043, 0.051] and its relative price weight -0.248 [-0.374, -0.123].

A.4 What the intervals measure

Behavioral intervals use 2,000 task-bootstrap resamples. The shuffled-price and no-price versions use the same resampled task indices, as do the checkpoints being compared, so intervals for changes and ratios preserve the dependence between their components. The separate listing-number tasks are resampled independently. Point ratios use full-sample coefficients and denominators, rather than the median bootstrap ratio.

These intervals measure how an estimate would vary with a different sample of tasks. They do not measure variation between neighboring checkpoints, between training runs or across prompt wordings, and neighboring checkpoints differ by more than task sampling predicts. For example, relative price weight is 0.37 [0.30, 0.44] at +13B and 0.09 [0.02, 0.16] at +17B. Across the five checkpoints from 3.880T to 3.896T its standard deviation is 0.14, against a mean task-bootstrap standard deviation of 0.04; across the nine annealing checkpoints of the tracked run the two values are 0.09 and 0.03. We summarize this excess with I^2, the share of the variation across checkpoints that task sampling does not explain. For relative price weight it is 0.90 and 0.86 in these two windows, and the price weight, decisiveness and the comparison slope show the same excess in these windows and across stage 1 from 3.146T (I^2 between 0.87 and 0.99). Within a training phase we therefore draw conclusions from patterns across many checkpoints, not from one step between neighbors. A post-training step, such as SFT to DPO, describes one run of the recipe, and runs also differ. Two DPO runs retrained on natural samples of the same preference data raise the plain-prompt relative price weight 1.45-fold and 2.84-fold (Appendix E.3). The untrained reference likewise describes the two tested initializations, not a population of random networks.

B Price-response fits and numerical readouts

This appendix gives the details behind Section 3. Appendix B.1 reports the fits behind the statement that the hotel-choice price response is closer to logarithmic than linear. Appendix B.2 describes how the internal number scale is measured and reports two checks on it.

B.1 Behavioral price-response fits

For each released stage and prompt format, we regress the 600 shuffled-price choice scores, without an intercept, on the six attribute differences and on the difference $f(p_{i1}) - f(p_{i2})$ of a price function f. The linear form uses $f(p) = p$ and the logarithmic form $f(p) = \log p$; each has one price coefficient. The flexible form is a step function with five bins at the quintiles of the 1,200 displayed prices, with the lowest bin as the reference, so it has four coefficients. The bin boundaries use only the displayed prices, not the scores. All three forms assume one additive price function for all hotels; the flexible form need not decrease with price.

Table A3 compares the forms in two ways. The main comparison is prediction error, the percentage reduction in mean squared error over 20 repetitions of five-fold cross-validation. Folds are drawn within the ten price-level groups of the shuffle (Appendix A.2) and are shared across stages, formats and forms, so differences between rows do not come from different splits. The repetitions reuse the same 600 tasks, so they are not independent replications. In the plain prompt, the logarithmic form has the lower error in all 20 repetitions at every released stage. The second comparison is the Akaike information criterion (AIC) of the Gaussian regression (Akaike, 1974); a positive linear-minus-log difference favors the logarithm. The AIC intervals and the quintile intervals in Figure 2 use 2,000 paired task-bootstrap draws within the same groups, refitting each form in every draw. They describe task variation conditional on these tasks and serve as descriptive diagnostics, not formal tests.

Table A3: **Behavioral price-form comparisons.** AIC differences are linear minus log, so positive values favor the logarithm; brackets give 95% task-bootstrap intervals. Prediction gains are percentage reductions in cross-validated mean squared error, positive when the first named form predicts better. The chat-template Base row is the format placebo of Appendix A.3.

Stage	$\mathrm{AIC}_{\mathrm{linear}} - \mathrm{AIC}_{\log}$ [95%]	Log vs. linear (%)	Bins vs. log (%)
Plain prompt			
Base	56.0 [35.8, 80.3]	8.8	−1.1
SFT	22.4 [12.5, 34.4]	3.6	1.5
DPO	65.4 [47.9, 84.0]	10.3	5.4
Instruct	61.6 [43.9, 80.3]	9.7	4.6
Chat template			
Base	−7.9 [−17.0, −0.4]	−1.3	−1.6
SFT	63.9 [47.1, 82.9]	10.1	5.6
DPO	79.8 [62.4, 98.8]	12.4	8.3
Instruct	66.3 [50.9, 83.8]	10.4	8.0

The five-bin form predicts better than the logarithm after post-training, by 1.5% to 8.3% at SFT, DPO and Instruct, but not at Base in the plain prompt (−1.1%), so the logarithmic form is an approximation. Adding the amenity and neighborhood indicators of the 71-control fit, or letting the six attribute slopes differ across the ten price-level groups, preserves the log-versus-linear directions. Both checks reuse the same 600 tasks.

Figure 2 plots the fitted price terms. Every regression uses all 1,200 displayed prices, which span \$51–\$1,624, but the figure draws lines only between the medians of the lowest and highest quintiles, \$150–\$662.5. Each dot is the estimate for a whole quintile, not for a single price, and the top quintile runs from about \$414 to \$1,624, so the figure says little about very high prices. Because the regression uses only price differences, the level of each curve is arbitrary. We therefore center each curve over the 1,200 prices, in every bootstrap draw as well, with bins weighted by their observed frequencies. Centering changes no fitted score or contrast. Because prices are shuffled only within price-level groups, price level remains associated with the hotel, so the fitted shape describes responses under this shuffle (Appendix A.2).

B.2 Numerical readout methods and checks

Section 3 reports three results about the internal number scale. The scale is closer to logarithmic than linear, untrained networks share this full-range spacing, and training adds magnitude information within equal digit counts by 399B. This subsection describes the number-pair grid, the readout methods and the power fit behind these results. It then checks whether the price exponent stays fixed across post-training and whether the untrained within-digit result depends on which numeral family is reported.

The readouts use a number-pair grid that is separate from the 303 two-price comparisons and the 303 bare-number comparisons in Table A1. It has 40 nonround values from 12 to 96,001, presented as prices, bare numbers, distances or populations. Each family's 780 unordered pairs appear in both orders. About 55% of the pairs form the fitting pool, and 20% of that pool is reserved for tuning a supervised readout; the other roughly 45% are held out. We call the remaining 80% of the pool the fitting pairs; the average, both directions and the regressions below use only these pairs. Both orders of a pair stay in the same set, and the split is the same at every checkpoint. Held-out pairs are new combinations of the same 40 values, so they test generalization to new pairs, not to unseen numbers.

We record the hidden state at the last prompt token after layers 14 and 19. A readout projects this state, minus its average over fitting pairs, onto one direction. The principal-component direction is the first principal component of the differences between the two orders of each fitting pair; numerical labels are used only to choose its sign. The difference-of-means direction joins the average hidden states of fitting pairs in the upper and lower thirds of the signed log ratio. Separate one-variable regressions, each with an intercept, predict the projection from the raw difference or from the log ratio, and held-out R^2 scores each on the held-out pairs. A supervised ridge readout fitted directly to

log ratios is excluded from the scale comparison, because its target already specifies the logarithm. Two layers and two directions are reported so that the scale result does not rest on one readout choice.

Figure A1a shows the comparison behind the statement in Section 3 that the logarithmic relation fits better in all 176 comparisons. They combine eleven checkpoints, four numeral families, two layers and two directions, and they share models and prompts, so they are not independent. The checkpoints are 3.608T, 3.754T and 3.896T in stage 1; +5B, +21B and +50B of the tracked run; the second run's endpoint; and Base, SFT, DPO and Instruct. The earliest is 3.608T, so this comparison does not show when the logarithmic spacing first appears.

The power fit places each readout on a scale from logarithmic to linear. We first assign one response value to each number, chosen so that differences between the values approximate the held-out pair projections. We then fit a Box–Cox power transformation of the numbers to these values (Box & Cox, 1964); its exponent is 0 for a logarithmic scale and 1 for a linear one. Exponents are searched from -1 to 2 in steps of 0.02. Their 95% intervals use 200 bootstrap samples of held-out pairs, keeping both orders together and the direction fixed. The fitted exponents range from -0.04 to 0.18. The untrained networks, measured on the price and bare-number prompts only, have exponents of -0.02 to 0 at both layers and with both directions, so the full-range logarithmic spacing does not by itself distinguish trained from untrained networks.

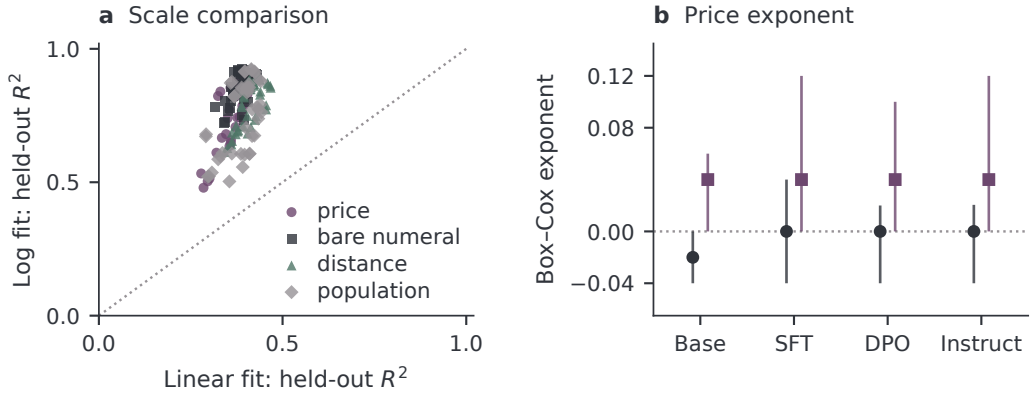


Figure A1: **Numerical scale fits.** (a) Held-out logarithmic and linear R^2 for the 176 comparisons, which share models and prompts; each point is one checkpoint, numeral family, layer and direction. The diagonal marks equal fit. (b) Price exponents at the last prompt token using the difference-of-means direction: gray circles show layer 14 and plum squares layer 19. Bars are 95% pair-bootstrap intervals conditional on the fitted direction. Exponent 0 is logarithmic; exponent 1 is linear and lies outside the displayed range.

Section 3 reports that the layer-19 difference-of-means price exponent is 0.04 at each released stage. At layer 14 it changes by 0.02 from Base to SFT, which is one step of the exponent search, and the intervals overlap (Figure A1b). The two annealing runs, which differ only in data order, have the same layer-14 exponent (-0.02) but exponents of 0.04 and 0.02 at layer 19, so runs of one recipe can differ by one step of the search. Exact stability across post-training holds only for the layer-19 difference-of-means readout, and only at the resolution of the search.

The within-digit test asks whether a readout carries magnitude information beyond digit count. It keeps each fitted direction and refits only the regression of the projection on the log ratio, using fitting pairs whose two numbers have the same number of digits. It then scores held-out pairs with the same number of digits, pooling digit lengths. Figure 3a and Section 3 report the weaker of the price and bare-number results at layer 19 with the difference-of-means direction. For the untrained networks the weaker family differs between the two tests. The weaker full-range values (0.69 and 0.71) come from the price prompts, and the weaker within-digit values (-0.07 and -0.12) from the bare-number prompts. The price prompts' within-digit values are also negative, so the untrained failure does not depend on this choice. Negative values mean that this readout fails the mean baseline.

They do not rule out information that another readout could recover, and the test does not isolate the contribution of tokenization. Every point in Figure 3a is computed with the same prompts, layer and readout procedure, so the procedure itself does not change along the series, although the plotted family can differ between checkpoints because each point is the weaker of the two. These R^2 values are point estimates without bootstrap intervals.

C PRETRAINING CHECKS

This appendix supports Section 4. Appendix C.1 asks whether the price weight rises during stage 1 before any division by decisiveness, and what the stage-1 choice scores measure. Appendix C.2 asks whether separation from the untrained networks depends on how decisiveness is measured. Appendix C.3 asks whether a comparison measure that does not saturate tracks the price weight. Appendix C.4 asks whether the fall in relative price weight at the start of annealing comes from the price weight or from decisiveness.

C.1 RAW PRICE WEIGHT AND ANSWER FORMAT

The raw price weight also rises during stage 1, so the rise in relative price weight is not only an effect of low decisiveness. Figure A2 shows the price weight w and decisiveness C separately. The raw weight rises from approximately zero at 399B to 0.035 at the stage-1 endpoint. Across the 15 stage-1 checkpoints, Kendall's τ between checkpoint order and the raw price weight is 0.73 (Spearman 0.91), and 0.73 again with 71 controls, while for decisiveness it is -0.24. The raw weight's interval lies above zero at 13 of the 15 checkpoints (all but 399B and 1.259T), including every checkpoint whose comparison accuracy is at least 90%. The rise is not monotone, and five of the 15 checkpoints fall in the last 16B stage-1 tokens, so we report τ descriptively.

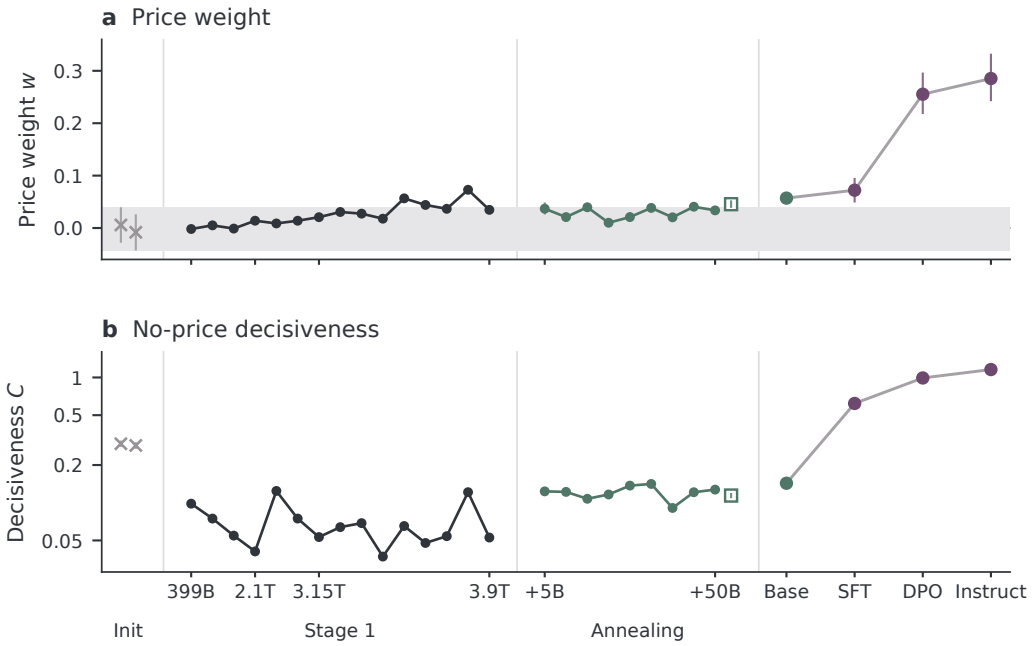


Figure A2: **Raw price weight and non-price decisiveness.** (a) Price weight w on the 600 shuffled-price tasks. (b) Decisiveness C on their no-price versions, with a logarithmic vertical axis. Figure 3c plots their ratio w/C. Bars are 95% paired task-bootstrap intervals in the plain prompt. Crosses mark untrained networks; the square marks the second annealing run. The gray band in a spans the untrained intervals. Horizontal spacing follows checkpoint order within each phase, not token counts.

Stage-1 choice scores are conditional on a letter answer. Table A4 gives the A/B mass, the mean probability on the A and B tokens, and slot-lock for every pretraining checkpoint and Base. From 2.937T to the end of stage 1 the A/B mass is 0.38 to 0.62, and six of these ten checkpoints have slot-lock above 0.95. Relative price weights at these checkpoints therefore measure shifts in the margin between A and B within a mostly fixed answer. At stage-1 checkpoints the higher-scored letter holds more probability than all tokens other than A and B combined on at most 3.0% of shuffled-price prompts, against 100% from +5B on. This condition is sufficient for greedy decoding to begin with that letter (Appendix D.2). Annealing raises the A/B mass above 0.72. Under the validity rule of Appendix A.3, 11 of the 15 stage-1 checkpoints would be excluded. The A/B mass is below 0.5 at nine and comparison accuracy below 0.90 at five, with three failing both, and only 3.880T, 3.889T, 3.893T and 3.896T pass. The untrained networks put almost no probability on A or B, so their band describes log-odds between two very unlikely tokens.

Table A4: **Answer-position diagnostics along the training trajectory.** Plain prompt, 600 shuffled-price hotel tasks. Slot-lock is the share of tasks on which the same answer letter wins in both option orders. A/B mass is the mean probability the model assigns to the two answer letters at the answer position.

Checkpoint	Slot-lock	A/B mass	Checkpoint	Slot-lock	A/B mass
399B	1.000	0.514	3.896T	0.960	0.565
848B	0.618	0.190	+5B	0.767	0.975
1.259T	1.000	0.214	+9B	1.000	0.924
2.098T	1.000	0.435	+13B	1.000	0.948
2.517T	0.368	0.491	+17B	0.998	0.869
2.937T	0.990	0.437	+21B	0.998	0.918
3.146T	1.000	0.494	+26B	0.997	0.774
3.356T	0.452	0.513	+34B	0.585	0.724
3.566T	0.993	0.380	+42B	0.512	0.775
3.775T	1.000	0.406	+50B	0.310	0.785
3.880T	0.435	0.625	+50B, second run	0.682	0.737
3.884T	1.000	0.421	Base	0.622	0.762
3.889T	0.818	0.584	Untrained 1	1.000	<0.001
3.893T	0.790	0.601	Untrained 2	0.847	<0.001

C.2 Sensitivity to normalization

This subsection asks whether the separation of the stage-1 relative price weight from the untrained band depends on how decisiveness is measured. Besides decisiveness on the no-price tasks, we use two other measures, each computed for every network. The first is the mean absolute choice score on the same 600 tasks at their original listed prices, which include prices and can therefore change when price sensitivity itself changes. The second is the mean absolute choice score on the 400 identical-product tasks that differ only in listing numbers, which measures responses to non-price numbers. A shared denominator instead divides every trained and untrained value by one constant, the stage-1 median no-price decisiveness (0.064 in the full sample, recomputed within each paired bootstrap draw). It rescales all networks alike and, in these data, identifies the same three checkpoints above the untrained band as the raw scale.

Table A5: **Pretraining sensitivity to normalization.** A checkpoint is above the untrained band when its entire 95% interval lies above both untrained networks' intervals. The last column gives the earliest stage-1 checkpoint from which every stage-1 checkpoint, including itself, is above the band. Own: each network's own measure; shared: one constant for all networks.

Scale	Stage-1 checkpoints above the band	Above from
Raw price weight	3 of 15	None
Own no-price decisiveness	10 of 15	3.146T
Shared no-price decisiveness	3 of 15	None
Own original-task decisiveness	10 of 15	3.146T
Own listing-number decisiveness	11 of 15	2.937T

Separation from the untrained band therefore depends on the normalization (Table A5). All three measures that use each network's own decisiveness separate the late stage-1 checkpoints from the band, whereas with the raw weight or the shared denominator only three late checkpoints lie above it. The separation of the relative weight in Figure 3c partly reflects lower decisiveness in the trained model. At the end of stage 1, $w = 0.035$ lies inside the raw untrained band $[-0.043, 0.040]$, but $C = 0.053$, against about 0.29 in the untrained networks, and the relative price weight, 0.66 [0.59, 0.74], lies well above the upper edge of its untrained band, 0.13. The untrained networks' larger decisiveness reflects arbitrary score differences, not better judgments about hotel quality. Decisiveness can also change as responses to other attributes change, so division by it does not isolate a pure price preference. The raw weight's interval lies above zero at 13 of the 15 stage-1 checkpoints (Appendix C.1) but above the untrained band at only three, so a positive price weight and separation from the untrained band are different criteria. The normalization changes both when separation begins and whether it persists.

C.3 Comparison slope

Comparison accuracy reaches 98.0% by 2.937T and stays near its ceiling at most later checkpoints (it dips to 85.1% at 3.356T and 91.7% at 3.884T), so it cannot show whether comparison and the price weight keep changing together. This subsection uses the comparison slope instead. It is the no-intercept least-squares slope of the order-averaged choice score on the 303 comparison questions, signed toward the cheaper price, on the absolute log price ratio; its interval resamples those questions. To compare the change in the slope between two checkpoints with the change in the raw price weight, we divide the slope ratio by the price-weight ratio and call the result the quotient. The quotient is 1 when the two measures change by the same factor. A common rescaling of all log-odds changes both by the same factor, so the quotient needs no division by decisiveness. Its interval combines paired draws of the comparison questions for the slope with paired draws of the hotel tasks for the price weight, because the two measures use different items.

The comparison slope and the raw price weight change by different factors at every step between neighboring checkpoints from 2.937T to +50B in the tracked run. The quotient's interval excludes 1 at all 18 steps. The quotient ranges from 0.27 [0.23, 0.31] (3.775T to 3.880T) to 3.74 [2.35, 11.13] (+13B to +17B), and the two measures sometimes move in opposite directions. Across the 26 trained pretraining checkpoints, the Spearman correlation between the comparison slope and the raw price weight is 0.37. For example, comparison accuracy is 98.0% at both 2.937T and 3.146T, while between them the raw price weight rises 1.49-fold [1.09, 2.39], relative price weight rises 2.09-fold [1.53, 3.35] and the comparison slope falls from 0.36 to 0.15. Over the same step the share of slot-locked comparison questions rises from 28% to 100%.

In the tracked run, the comparison slope rises at the start of annealing while the price weight shows no detectable change. From 3.896T to +5B the slope rises 3.41-fold [3.27, 3.57], from 0.56 [0.52, 0.61] to 1.92 [1.76, 2.13], while the price weight changes 1.06-fold [0.76, 1.40], so the quotient is 3.21 [2.44, 4.43]. Every annealing checkpoint of the tracked run has a higher slope than every stage-1 checkpoint from 3.146T, and all 81 paired intervals exclude 1. The second annealing run does not share the rise; its slope at +50B, 0.49 [0.45, 0.54], lies within the stage-1 range. The rise also coincides with a change in answer format. On comparison questions the probability on A or B rises from 0.62 to 0.99 over the same step (Table A4 gives the hotel tasks). A common rescaling of log-odds leaves the quotient unchanged, but a change in answer format need not be a common rescaling, so the slope rise need not reflect better comparison alone. Unlike the comparison slope, the within-digit held-out R^2 plotted in Figure 3a (Appendix B.2), which has no saved intervals, rises in both annealing runs. It is higher at every annealing checkpoint (0.65 to 0.74, including the second run and Base) than at any stage-1 checkpoint (at most 0.58).

C.4 Changes during annealing

This subsection asks whether the fall in relative price weight at the start of annealing comes from the price weight or from decisiveness. Over the first 5B annealing tokens (3.896T to +5B), decisiveness rises from 0.053 to 0.123, while the paired change in the raw price weight is $+0.002$ $[-0.008, +0.014]$. In the six-control fits to the no-price choice scores, the star-rating coefficient rises from 0.0105 to 0.0495 and the breakfast coefficient from 0.0170 to 0.0635. These coefficients are point

estimates without saved intervals. They grow by different factors, so these fits do not show one common factor rescaling all answers; their rise is consistent with stronger responses to hotel quality cues.

The rise in decisiveness does not depend on starting the comparison at 3.896T. Measured from 3.884T or 3.889T instead, decisiveness at +5B is 2.6 or 2.3 times as large, while the price weight changes by at most 17%. Near the end of stage 1, one checkpoint, 3.893T, has decisiveness near the annealing level (0.121, against 0.053 at 3.896T), but its price weight is also about twice its neighbors' (0.073), so its relative price weight barely moves (0.60, against 0.68 at 3.889T and 0.66 at 3.896T). At the start of annealing, in contrast, decisiveness rises while the price weight does not.

Neighboring checkpoints differ by more than their task-bootstrap intervals predict (Appendix A.4), so whole-phase comparisons are more informative than any single step, and they give the same result. Median decisiveness is 0.122 across the nine annealing checkpoints of the tracked run and 0.054 across the nine stage-1 checkpoints from 3.146T, while the median price weights are 0.034 and 0.035. Relative price weight lies between 0.09 and 0.37 at every annealing checkpoint of the tracked run and between 0.39 and 0.92 at every stage-1 checkpoint from 3.146T, and 78 of the 81 paired comparisons have intervals excluding 1.

These comparisons do not isolate an effect of the annealing mixture. The two annealing runs use the same data in different orders and both receive 50B additional tokens, so their difference describes variation under that recipe, not a comparison with continued stage-1 training. Base averages the three runs' model weights, which need not average their measured price preferences.

D Post-training checks

This appendix supports Section 5. Appendix D.1 asks whether its conclusions depend on what the price weight is divided by. Appendix D.2 explains how price can be measured when one answer letter usually wins, and when the choice scores describe the answer a model would generate. Appendix D.3 reruns the weighting study on five other models to separate the patterns that recur from those specific to OLMo 2 7B. Appendix D.4 asks whether the post-training changes extend to other numerical attributes.

D.1 Other normalizations of the price weight

The relative price weight divides the price weight w by decisiveness C, the mean absolute choice score on the same tasks with prices removed. A different denominator could change the conclusions of Section 5, so we repeat its paired comparisons with three alternatives. Two divide w by other mean absolute choice scores. The first uses the same 600 tasks at their original listed prices, which include prices, so this score can change with the price response itself. The second uses the 400 identical-product tasks that differ only in a listing number (Appendix C.2). The third needs no measure of decisiveness. It divides w by each quality coefficient from the same fit and takes the geometric mean of these ratios over star rating, free cancellation, breakfast and review score. This price-to-quality ratio is the inverse of the willingness to pay that conjoint analysis computes, and any common scale cancels in it. Table A6 gives each change as the ratio of the later to the earlier checkpoint, with paired task-bootstrap intervals (Appendix A.4).

For the OLMo 2 7B Base-to-SFT and SFT-to-DPO steps, every normalization gives the same direction. In the plain prompt SFT lowers the relative price weight and DPO raises it, and DPO also raises it in the chat template. For these steps only the size of the change depends on the normalization. The price-to-quality ratio also moves in the same direction as the main measure at the start of annealing (3.896T to +5B), where it changes 0.39-fold [0.27, 0.52] against 0.45-fold. Single attributes can disagree. From Base to DPO the ratio for star rating rises 1.38-fold while the other three fall. In the plain-prompt DPO step of Tulu 3 8B the price-to-quality ratio and the main measure move in opposite directions, so their agreement need not hold in other models.

Whether DPO ends below the pretraining levels depends on the normalization (last two rows of Table A6). Under the main measure its relative price weight is below both the Base value and the value at the end of stage 1 (3.896T). Divided by the original-task scores, it is below the stage-1 value, and its comparison with Base is inconclusive. Divided by the listing-number scores, it is above both.

Table A6: **Post-training changes under four normalizations.** Each column divides the same shuffled-price weight w by a different quantity; entries are later/earlier ratios with 95% paired task-bootstrap intervals. No-price C is the main measure. Price-to-quality: the geometric mean of the ratios of w to the star-rating, cancellation, breakfast and review-score coefficients. —: not computed. The changes in w and C for the first three rows are in Table A9; from Base to DPO, w rises 4.47-fold [3.84, 5.28] and C 6.93-fold [6.69, 7.18].

		w divided by			
Step	Format	No-price C	Original tasks	Listing numbers	Price-to-quality
Base → SFT	Plain	0.29 [0.22, 0.36]	0.34 [0.26, 0.43]	0.47 [0.34, 0.59]	0.27 [0.20, 0.34]
SFT → DPO	Plain	2.21 [1.90, 2.83]	2.50 [2.14, 3.18]	2.71 [2.31, 3.48]	2.34 [2.01, 3.02]
SFT → DPO	Chat	1.07 [1.02, 1.13]	1.26 [1.20, 1.34]	1.23 [1.14, 1.32]	1.24 [1.18, 1.31]
Base → DPO	Plain	0.65 [0.56, 0.76]	0.86 [0.74, 1.01]	1.27 [1.07, 1.54]	0.63 [0.53, 0.75]
3.896T → DPO	Plain	0.39 [0.33, 0.46]	0.40 [0.34, 0.46]	1.52 [1.26, 1.82]	—

The statement in Section 5 that DPO stays below both pretraining levels therefore holds for the main measure only.

D.2 Answer position and greedy decoding

After post-training, one answer letter often wins in both option orders (slot-lock; Section 5). This subsection explains why price can still be measured under slot-lock, and when the choice scores describe the answer a model would generate.

Each order's log-odds has a part shared by both orders and a part that follows the hotels. The shared part is the position score $b_i = (\ell_i^{AB} + \ell_i^{BA})/2$, and the part that follows the hotels is the choice score d_i of Section 2, so $\ell_i^{AB} = b_i + d_i$ and $\ell_i^{BA} = b_i - d_i$. With no ties, the same letter wins in both orders exactly when $|b_i| > |d_i|$. Slot-lock therefore means that the position preference is larger than the hotel difference, not that the hotel difference is zero. Price still moves d_i, and w measures that movement. On the 600 shuffled-price tasks in the chat template, DPO lowers the mean $|b_i|$ from 1.561 to 0.927 and raises the mean $|d_i|$ from 0.671 to 1.069, consistent with the fall in slot-lock from 96.8% to 45.2% (Table 2). In the plain prompt both grow, from 1.478 to 1.601 and from 0.514 to 0.797, and slot-lock remains high.

Choice scores compare A with B only, but a model can also put probability on other tokens. Greedy decoding begins with the higher-scored letter whenever that letter holds more probability than all tokens other than A and B combined. In the plain prompt at Base, SFT, DPO and Instruct, this holds on every one of the 1,200 shuffled-price and 1,200 no-price prompts. In the chat template it holds on 100% of prompts at SFT and Instruct and on 98.6% at DPO, where all exceptions are no-price prompts. During pretraining it holds on every shuffled-price prompt from +5B on, but rarely at stage 1 (Appendix C.1). It is not universal across models. Among the valid measurements in Appendix D.3, the share of prompts on which it holds ranges from 0% (Zephyr 7B SFT, plain prompt) to 100%. Whether choice scores describe generated answers must therefore be checked for each model and format.

D.3 Other models

Why rerun the study. Section 5 follows one model, so its patterns could reflect the OLMo 2 recipe rather than post-training in general. The weighting study needs only a model's released checkpoints, so we reran it unchanged on five other models (Table A7): the same 4,006 items (Table A1), the same scoring code, the plain prompt and each model's own chat template, with inputs tokenized as in each model's training. For OLMo 2 7B and OLMo 2 13B we also use two variants of the chat template. The cue format places the question in the user turn and starts the reply with the plain prompt's "Answer:" cue. The system-message format adds "You are a helpful assistant." to the OLMo 2 chat template. The rerun separates the patterns that recur from those specific to OLMo 2 7B, and it runs the audit template on models it was not built for. We score only their released Base and post-training checkpoints, so these models show post-training only.

Table A7: **Models in the cross-model rerun.** Released checkpoints on Hugging Face. The Llama 3.1 8B and Llama 3 8B base models require access approval and were not scored. The OLMo 2 1B (`allenai/OLMo-2-0425-1B`) and Mistral 7B (`mistralai/Mistral-7B-v0.1`) base models were scored but fail the comparison condition in both formats.

Model	Stages	Repositories
OLMo 2 13B (Team OLMo et al., 2025)	Base, SFT, DPO	`allenai/OLMo-2-1124-13B`, `-SFT`, `-DPO`
Olmo 3 7B (Team Olmo et al., 2025)	Base, SFT, DPO	`allenai/Olmo-3-1025-7B`, `allenai/Olmo-3-7B-Instruct-SFT`, `-DPO`
Tulu 3 8B (Lambert et al., 2025), trained from Llama 3.1 8B (Grattafiori et al., 2024)	SFT, DPO	`allenai/Llama-3.1-Tulu-3-8B-SFT`, `-DPO`
Llama 3 8B SFT and DPO (Meng et al., 2024), trained from the Llama 3 8B base model	SFT, DPO	`princeton-nlp/Llama-3-Base-8B-SFT`, `-SFT-DPO`
Zephyr 7B (Tunstall et al., 2024), trained from Mistral 7B (Jiang et al., 2023)	SFT, DPO	`HuggingFaceH4/mistral-7b-sft-beta`, `HuggingFaceH4/zephyr-7b-beta`

Which measurements count. We use only valid measurements (Appendix A.3), in which the checkpoint answers with a letter and compares prices correctly. Base models ship no chat template, so each is scored with its SFT checkpoint's template, where every Base model puts at most 37% of its probability on A or B and is excluded. Table A8 marks the excluded measurements of the listed checkpoints; all fall short of the A/B-mass condition. A change between two checkpoints is valid when both measurements are, and Table A9 gives the valid changes. The release of Meng et al. (2024) also includes eight checkpoints trained from the same SFT checkpoint with other preference objectives. We scored them only in the chat template, where that SFT checkpoint is not valid, so they give levels, not changes. Counting the cue and system-message formats of OLMo 2 13B and these objectives, 27 measurements of the other models are valid. Levels differ widely between models, so we compare changes.

Table A8: **Validity and relative price weight of other models.** Released checkpoints are scored on the same 600 shuffled-price tasks as Table 2, in the plain prompt and in each model's own chat template (Base models use their SFT checkpoint's template). Each $r = w/C$ has a 95% task-bootstrap interval. A/B mass is averaged over the 1,800 original tasks, where the validity rule applies; slot-lock uses the 600 shuffled-price tasks, as in Table 2. Table 2 gives the OLMo 2 7B levels. —: excluded because A/B mass is below 0.5. Levels differ widely between models, so the text compares the changes in Table A9.

		Plain prompt			Chat template		
Model	Stage	r [95%]	A/B mass	Slot-lock	r [95%]	A/B mass	Slot-lock
OLMo 2 13B	Base	2.64 [2.41, 2.92]	0.98	0.30	—	0.37	1.00
	SFT	1.21 [1.11, 1.32]	0.99	0.11	1.49 [1.38, 1.63]	0.90	0.13
	DPO	1.35 [1.25, 1.47]	1.00	0.11	1.64 [1.51, 1.80]	0.89	0.19
Olmo 3 7B	Base	0.88 [0.77, 0.99]	0.86	0.35	—	0.01	1.00
	SFT	—	0.41	0.46	1.65 [1.53, 1.79]	0.99	0.13
	DPO	—	0.47	0.28	1.21 [1.12, 1.32]	1.00	0.19
Tulu 3 8B	SFT	0.59 [0.54, 0.64]	0.96	1.00	0.53 [0.49, 0.57]	0.76	0.69
	DPO	0.57 [0.52, 0.62]	0.99	0.98	0.54 [0.50, 0.59]	0.54	0.78
Llama 3 8B	SFT	0.85 [0.78, 0.94]	0.57	0.75	—	0.15	0.95
	DPO	1.29 [1.18, 1.41]	0.95	0.56	0.54 [0.48, 0.60]	0.98	0.70
Zephyr 7B	SFT	0.26 [0.20, 0.32]	0.56	0.83	—	0.23	1.00
	DPO	—	0.16	0.40	—	0.00	0.64

Table A9: **Paired changes across released stages in OLMo 2 7B and four other models.** Entries are later/earlier ratios with 95% task-bootstrap intervals. Zephyr 7B is omitted because its DPO checkpoint is excluded in both formats. The cue format places the question in the chat template's user turn and starts the reply with the plain prompt's "Answer:" cue; the system-message format adds "You are a helpful assistant." to the OLMo 2 chat template. —: an endpoint puts less than half of its answer probability on A or B. ‡An endpoint has slot-lock above 0.90 on the shuffled-price tasks.

Model	Step	Price weight w	Decisiveness C	Relative price weight r
Plain prompt				
OLMo 2 7B	Base → SFT	1.26 [0.95, 1.57]	4.33 [4.21, 4.46]	0.29 [0.22, 0.36]‡
OLMo 2 7B	SFT → DPO	3.54 [3.04, 4.52]	1.60 [1.58, 1.62]	2.21 [1.90, 2.83]‡
OLMo 2 13B	Base → SFT	0.83 [0.81, 0.86]	1.83 [1.78, 1.87]	0.46 [0.44, 0.47]
OLMo 2 13B	SFT → DPO	2.67 [2.64, 2.69]	2.39 [2.37, 2.40]	1.12 [1.10, 1.13]
Olmo 3 7B	SFT → DPO	—	—	—
Tulu 3 8B	SFT → DPO	2.31 [2.27, 2.34]	2.39 [2.37, 2.42]	0.96 [0.95, 0.98]‡
Llama 3 8B	SFT → DPO	9.37 [9.11, 9.66]	6.23 [6.13, 6.33]	1.50 [1.46, 1.56]
Chat template				
OLMo 2 7B	SFT → DPO	1.91 [1.83, 2.02]	1.79 [1.77, 1.81]	1.07 [1.02, 1.13]‡
OLMo 2 13B	SFT → DPO	2.79 [2.77, 2.82]	2.54 [2.52, 2.56]	1.10 [1.08, 1.11]
Olmo 3 7B	SFT → DPO	0.96 [0.95, 0.98]	1.32 [1.30, 1.33]	0.73 [0.72, 0.75]
Tulu 3 8B	SFT → DPO	1.99 [1.96, 2.03]	1.93 [1.91, 1.95]	1.03 [1.01, 1.05]
Llama 3 8B	SFT → DPO	—	—	—
Cue format				
OLMo 2 7B	Base → SFT	4.13 [3.62, 4.72]	7.41 [7.14, 7.71]	0.56 [0.49, 0.64]‡
OLMo 2 7B	SFT → DPO	1.68 [1.60, 1.78]	1.62 [1.60, 1.64]	1.04 [0.99, 1.10]‡
OLMo 2 13B	Base → SFT	6.03 [5.70, 6.40]	5.09 [4.97, 5.22]	1.18 [1.11, 1.26]
OLMo 2 13B	SFT → DPO	3.44 [3.41, 3.46]	2.48 [2.46, 2.51]	1.38 [1.37, 1.40]
System-message format				
OLMo 2 7B	SFT → DPO	2.96 [2.68, 3.44]	1.91 [1.88, 1.93]	1.55 [1.41, 1.81]‡
OLMo 2 13B	SFT → DPO	3.45 [3.40, 3.50]	3.16 [3.13, 3.18]	1.09 [1.07, 1.11]

What recurs. Four patterns recur. The first two hold in every valid measurement, the third has one exception, and the fourth holds across models and objectives but not across runs of one recipe. First, the price weight favors the cheaper option before post-training. In every valid base-model measurement it is positive, with an interval above zero. This holds for OLMo 2 13B and Olmo 3 7B in the plain prompt (Table A8), as for OLMo 2 7B Base, and for both OLMo 2 base models in the cue format. The Mistral 7B and OLMo 2 1B base models fail the comparison condition, so the pattern concerns models that compare prices accurately.

Second, SFT and DPO raise decisiveness. All 24 valid SFT and DPO steps raise it, each with an interval excluding 1, by 1.32-fold (Olmo 3 7B, DPO, chat template) to 7.41-fold (OLMo 2 7B SFT, cue format). The 24 steps are the 16 valid rows of Table A9 and the eight changes of Table A11, four retraining runs (Appendix E.3) in two formats. They are not 24 independent models. The one other post-training step, reinforcement learning from DPO to Instruct in OLMo 2 7B, lowers decisiveness in the chat template (0.83 [0.82, 0.84]).

Third, DPO raises the raw price weight, in 19 of the 20 valid DPO steps (12 in Table A9 and 8 in Table A11); the exception is Olmo 3 7B in the chat template (0.96). Because decisiveness rises as well, a raw attribute weight mixes preference with sharpness.

Fourth, dividing by decisiveness narrows differences between models and objectives, because the price weight and decisiveness tend to rise together. Across the eight valid DPO steps of OLMo 2 7B and the four other models in the plain prompt and chat template, the standard deviation of the log change is 0.64 for the price weight and 0.33 for the relative price weight. The preference objectives of Meng et al. (2024), all trained from one SFT checkpoint, give a sharper test. The seven that are valid (DPO and six others) differ 5.94-fold [5.68, 6.18] in price weight and 7.60-fold in decisiveness, but only 1.53-fold [1.48, 1.60] in relative price weight. Normalization does not remove variation

between runs of one recipe, however. Across the four retraining runs, decisiveness rises by similar factors (1.65- to 1.94-fold in the plain prompt), so the relative price weight (1.45- to 2.84-fold) varies about as much as the price weight (2.70- to 5.53-fold).

What varies. The size of the DPO step in relative price weight does not recur. In the plain prompt it is 2.21-fold in OLMo 2 7B and 0.96- to 1.50-fold in the other models, so the OLMo 2 7B plain-prompt increase is the largest DPO change among the released checkpoints, not a typical one. In each model's own chat template it is 0.73- to 1.10-fold, a range that includes the OLMo 2 7B value of 1.07. The chat templates also differ. The Olmo 3 template adds a default system message about function calling, and the others add none. A system message alone can change the step. In the system-message format the OLMo 2 7B step is 1.55-fold [1.41, 1.81], while the 13B step is 1.09-fold [1.07, 1.11]. The prompt format matters too. In the cue format, which moves the plain prompt's "Answer:" cue into the chat template, the OLMo 2 7B step is not detectable (1.04 [0.99, 1.10]). The retraining runs show that the step also varies between runs of one recipe (Appendix E.3). An audit of a post-training step therefore needs to name the model, the format and any system message.

SFT. Only OLMo 2 7B and OLMo 2 13B have valid Base and SFT measurements in the same format. In OLMo 2 13B, SFT also lowers the relative price weight in the plain prompt, to 0.46 [0.44, 0.47] times its Base value. As in OLMo 2 7B, the decline comes mainly from decisiveness, which rises 1.83-fold, while the price weight falls to 0.83 times its Base value. The 13B SFT checkpoint is slot-locked on only 11% of the shuffled-price tasks, against 99.7% for OLMo 2 7B, so the decline does not require slot-lock. It does depend on the prompt. Both Base models are valid in the cue format, and there the 13B relative price weight rises 1.18-fold [1.11, 1.26] from Base to SFT, while the OLMo 2 7B value falls to 0.56 [0.49, 0.64] times its Base value. In that format the decline appears only in OLMo 2 7B. Its SFT checkpoint is slot-locked on 99.7% of the tasks, against 42% for 13B, so these two models cannot separate the effect of the format from that of slot-lock. The SFT decline therefore recurs in the plain prompt under the main measure, but not in every format. In the plain prompt of OLMo 2 13B it also reverses under the original-task normalization (1.29 [1.23, 1.35]).

D.4 Responses to other numerical attributes

This check asks whether post-training also changes how numerical attributes other than price influence choices. If it does, the price results are part of a broader change. It is the behavioral counterpart of the text audit of the same seven quantities in Appendix E.1.

There are 400 tasks for each of seven attributes: price, distance from the city centre, rating count, listing number and district population in hotel choices; mileage and model year in used-car choices. We score both option orders in the plain prompt at 2.098T, 3.775T, Base, SFT, DPO and Instruct, and in the two untrained networks. Target values are assigned independently of product identity; the other numerical slots are held equal. Quality attributes still differ and enter the regression as controls: stars, review score, bed-size rank, cancellation, breakfast and amenity count for hotels; inspection grade, dealer review score, review count, warranty and certification for cars. We regress the order-averaged choice score on these control differences and the target log-value difference divided by its standard deviation, without an intercept. The resulting coefficient measures sensitivity over each attribute's numerical range and is not divided by decisiveness.

At SFT, DPO and Instruct, the responses favor cheaper prices, shorter distances and larger populations, with 95% task-bootstrap intervals wholly outside each attribute's untrained reference, the span of the two untrained networks' intervals. Mileage, rating count, model year and listing number overlap their references at every tested checkpoint. At the two stage-1 checkpoints and Base, all seven attributes overlap their references.

The changes after post-training therefore extend beyond price. They are visible from SFT on, so they are not specific to DPO. They are not evidence for one change shared by all attributes, because four of the seven never leave their reference. These coefficients come from separate tasks and are not divided by decisiveness, so they do not measure the price preference itself. Their overlap with the untrained references during pretraining does not contradict the main result, which concerns the relative price weight.

E TRAINING-DATA AUDITS AND DPO RETRAINING

This appendix reports two training-data audits and one DPO retraining experiment. Appendix E.1 asks what share of genuine price mentions in pretraining web text occur in comparisons that help settle a choice, and whether that share is larger than for other quantities. Appendix E.2 asks whether the responses that DPO's labels prefer favor cheaper options more often than the responses they reject. Appendix E.3 asks whether DPO still raises the relative price weight when price-flagged pairs are replaced in its training data.

E.1 CORPUS EXPOSURE IN PRETRAINING TEXT

We sampled 150,000 documents from the DataComp for Language Models (DCLM) web-text component (Li et al., 2024) of each pretraining stage (OLMo-Mix-1124 for stage 1, Dolmino-Mix-1124 for annealing), using a seeded streaming shuffle. Because only this component is sampled, the samples do not represent the full training mixtures. The audit covers prices of goods and services generally, not only hotels, and six other quantities: distance, rating count, listing number, population, mileage and model year, the attributes of the check in Appendix D.4. A lexical screen identified candidate mentions; Table A10 shows how many price windows were sampled and judged genuine.

Table A10: **Training-data audit samples.** Top: candidate price windows found by the lexical screen in each 150,000-document DCLM sample. Across all seven quantities, 2,176 windows were judged, 1,097 from stage 1 and 1,079 from annealing. Bottom: flagged preference pairs by sampling group, drawn from 378,341 pairs.

Sample	Candidates	Judged	Genuine prices
Pretraining price windows (Appendix E.1)			
Stage-1 DCLM	75,790	200	87
Annealing DCLM	50,723	200	73
Preference pairs (Appendix E.2)			
Both responses flagged, comparison or recommendation cue	6,641	800	—
Both responses flagged, no such cue	11,564	500	—
One response flagged	27,310	300	—

For each corpus, we drew 50 price windows from each of four groups defined by the screen's two predictions, whether the passage compares alternatives and whether it settles a choice. A window extends 240 characters on either side of the mention. One LLM judge, Anthropic's Claude Fable 5 (queried in August 2026), assessed whether the number was a genuine instance of the quantity, for example a genuine price, whether alternatives were compared, and whether the passage helped settle a choice. Choosing an item because it costs less meets the last two conditions. Insufficient context was coded false. This audit has no second judge, unlike the preference audit, and neither audit is validated against human labels.

Equal sampling across groups overrepresents rarer contexts. We therefore weight each group by its candidate population and divide the weighted count meeting all three conditions by the weighted count of genuine prices. Windows for the other six quantities were drawn within the same four groups and weighted in the same way. In stage-1 text this gives 31.8% [19.0, 46.7], the estimate in Section 4, against an unweighted 41/87 (47.1%). Intervals use 2,000 resamples of windows within groups, recomputing both counts. They reflect only the uncertainty from which windows were judged, and they are conditional on the sampled documents, the screen and the judge. They omit missed mentions, judge error and dependence among mentions in the same document.

A larger share of genuine price mentions than of any other measurable quantity's mentions occurs in comparisons that help settle a choice, in both corpora, except model year in stage-1 text, where the difference is inconclusive. A quantity is measurable in a corpus if it has at least 30 genuine mentions there. In stage-1 text (Figure A3), the price share exceeds those of distance, population, rating count and listing number, each difference with a 95% interval that excludes zero, and its difference from model year is 16.6 percentage points [−3.9, 34.8]; mileage is not measurable there. In annealing text, the price share is 16.2% [7.0, 28.2] and exceeds the four measurable comparators, while mileage and model year are not measurable. The seven quantities are a selected comparison

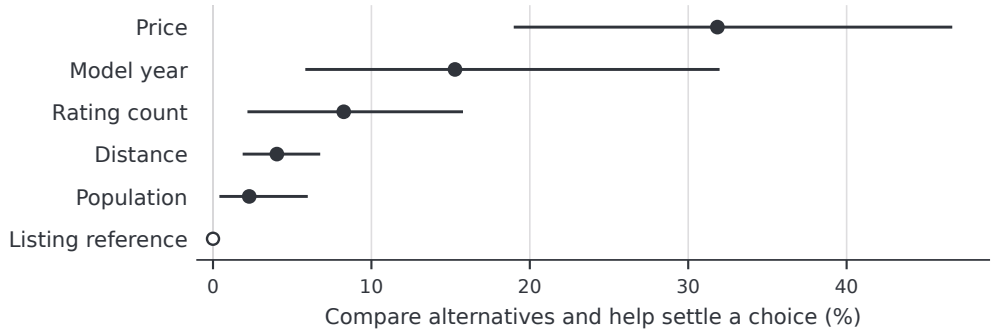


Figure A3: **Choice-relevant numerical contexts.** Weighted share of genuine mentions that compare alternatives and help settle a choice in stage-1 DCLM text. Bars show 95% intervals from resampling judged windows within sampling groups. The open marker is an observed zero, not evidence that the population rate is zero.

set, not a representative sample of numerical concepts. The judge did not code which option a comparison favored, so these contexts identify possible learning signals without showing that they teach a preference for cheaper options or cause the observed model behavior.

E.2 Preference labels

This audit asks whether the responses that DPO's labels prefer (chosen responses) favor cheaper options more often than the responses they reject (rejected responses). The labels come from the 378,341-pair public preference set. After code and mathematics are masked, a lexical screen flags 45,515 pairs in which a response contains a currency amount or price, cost or value language; the screen does not consult which response the label prefers. We sample randomly within the three groups in Table A10, deliberately emphasizing comparisons and recommendations, so the 1,600 pairs are an enriched sample, not a uniform sample of the preference set.

The primary judge, Claude Sonnet 5 (`claude-sonnet-5`, queried 19 August 2026), gives each response one of four labels: favors the cheaper option, favors the costlier option, contains price content without a direction, or contains no price content. Each pair is presented twice, once in each response order, and the judge is not told which response was chosen. Excerpts are capped at 1,500 characters; 1,250 of the 1,600 pairs are truncated. A second judge, Claude Opus 5 (`claude-opus-5`), agrees with the primary judge on 88.5% of 156 response labels from 78 pairs. Most of these labels are nondirectional, so the agreement mainly concerns them, and it is not a human validation.

Cheaper-favoring language appears in 13.19% of chosen and 13.03% of rejected responses, a paired difference of 0.16 percentage points $[-0.88, 1.19]$ (Section 5). Each rate averages each pair's two presentations and is not reweighted by sampling group, so the rates describe the selected sample, not the full preference set. The interval uses 4,000 bootstrap resamples of pairs, keeping each pair's two responses together.

Few pairs contrast price directions directly. Only one pair has a cheaper-favoring and a costlier-favoring response in both presentation orders. A broader contrast needs one cheaper-favoring response and one response with any other label. Fifty-one pairs qualify in both orders, and in 48.0% [35.3, 61.8] of them the cheaper-favoring response is the chosen one. The interval is wide, so the audit has limited power. It detects no directional imbalance in these pairs, but it cannot exclude weaker effects, indirect signals or content that the screen or the excerpts missed.

E.3 DPO retraining with and without price-flagged pairs

This experiment asks whether DPO still raises the relative price weight when the pairs flagged by a price screen (price-flagged pairs, the price-related pairs of Section 5) are replaced by unflagged pairs. The replacement lowers but does not remove price content. We retrained DPO from the released SFT checkpoint of OLMo 2 7B with the published settings, for 500 steps per run, each on a 64,000-pair

sample of the 378,341-pair preference data. Two runs used a natural sample and two a sample in which price-flagged pairs were replaced by matched unflagged pairs, with seeds 42 and 43 in each arm. Table A11 gives each run's change from SFT.

Four results hold in all four runs. The price weight rises in both formats (1.99- to 5.53-fold). Relative price weight rises in the plain prompt, including in both runs without price-flagged pairs. Decisiveness rises in both formats. The chat-template change in relative price weight is smaller than the plain-prompt change, and in one run it is not detectable. The size of the step varies between runs. The plain-prompt rise is 1.45-fold with one natural-sample seed and 2.84-fold with the other, and their intervals do not overlap, so in these smaller runs the size of a single DPO step is uncertain well beyond task sampling. The comparison between arms reverses with the seed. With seed 42 the run without price-flagged pairs has the larger plain-prompt rise (1.90 against 1.45), and with seed 43 the smaller (2.17 against 2.84), so two seeds per arm cannot show whether the price-flagged pairs matter. We stopped after two seeds per arm, after these results had been viewed.

Table A11: **DPO retraining runs.** Changes from the released SFT checkpoint of OLMo 2 7B, as later/earlier ratios with 95% paired task-bootstrap intervals, for the released DPO checkpoint (repeated from Table A9) and four DPO runs retrained with the published settings on 64,000-pair samples of its preference data. Every row shares the SFT endpoint, whose slot-lock is above 0.90 in both formats (Table 2).

Run	Price weight w	Decisiveness C	Relative price weight r
Plain prompt			
Released DPO	3.54 [3.04, 4.52]	1.60 [1.58, 1.62]	2.21 [1.90, 2.83]
Natural sample, seed 42	2.70 [2.39, 3.25]	1.86 [1.83, 1.88]	1.45 [1.29, 1.75]
Natural sample, seed 43	5.53 [4.66, 7.24]	1.94 [1.92, 1.97]	2.84 [2.39, 3.73]
Price-flagged replaced, seed 42	3.40 [2.97, 4.22]	1.79 [1.77, 1.82]	1.90 [1.65, 2.37]
Price-flagged replaced, seed 43	3.58 [3.08, 4.57]	1.65 [1.63, 1.68]	2.17 [1.87, 2.77]
Chat template			
Released DPO	1.91 [1.83, 2.02]	1.79 [1.77, 1.81]	1.07 [1.02, 1.13]
Natural sample, seed 42	1.99 [1.88, 2.12]	2.03 [2.01, 2.05]	0.98 [0.93, 1.05]
Natural sample, seed 43	2.88 [2.68, 3.19]	1.51 [1.49, 1.52]	1.91 [1.77, 2.11]
Price-flagged replaced, seed 42	2.19 [2.06, 2.35]	1.81 [1.79, 1.83]	1.21 [1.14, 1.30]
Price-flagged replaced, seed 43	2.12 [1.99, 2.31]	1.50 [1.48, 1.51]	1.42 [1.32, 1.55]