DECIDE, DON'T DECODE: TURNING POST-TRAINING LANGUAGE MODELS INTO CALIBRATED, COHERENT DECISION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many systems use language models for bounded decisions, such as routing a ticket or flagging a message, and act on the model's confidence. We develop a post-training method that turns an open generative model into a typed decision model: for a choice, a rating or a yes/no question it returns probabilities over the allowed answers without decoding text, trained to be calibrated and coherent across related questions. We build DECIDE-22, 22 typed decision tasks with 22,773 test items; on it, decoding the answer with a stated confidence is no more accurate than reading these probabilities, costs several times more, and yields a confidence that barely separates right from wrong answers (AUROC 0.59, and 0.69 after reasoning, against 0.76). Tracing 26 checkpoints of six model families through every post-training stage they publish, we find that preference optimization sharpens the supervised readout into over-confidence at unchanged accuracy, almost exactly as a temperature would (R^2 0.92–0.95 in three pipelines). Fine-tuning the readout with a proper scoring rule reverses this sharpening and improves accuracy and calibration, but increases contradictions across logically related questions. We address this with a penalty based on the de Finetti sure loss of each question and its automatically derived siblings. The penalty needs no additional labels and reduces incoherence tenfold without a measurable loss in accuracy. A 9B open model trained this way reaches 76.0% accuracy and a calibration error of 0.054, against 73.3% and 0.113 for a commercial decision API as served (0.068 when given the same calibration recipe), with under a third of its incoherence; at 4B it also beats a concurrent open decision model trained on six times the data. The API keeps its lead on the six held-out tasks, most of it from a single rating scale. We release the benchmark, code and trained models.

1 INTRODUCTION

A growing share of calls to language models do not ask for text. A support system asks which queue a ticket belongs in; a moderation pipeline asks whether a message is abusive; an agent asks whether a tool call is safe to run. The answer is one of a few allowed options, and what the system does next depends as much on the model's confidence as on its choice: act, escalate to a person, or ask for more information. For calls like these the useful output is a probability distribution over the allowed answers, and it is only useful if it can be trusted.

Commercial providers now sell exactly this. Jev answers typed questions (pick one of K options, place an item on an ordered scale, or say whether a statement holds) with a probability distribution and no generated text (TypeSafe, 2026); its method and training are not public. Open models can be asked the same questions in two ways. They can decode an answer, which is slow and yields a verbalized confidence loosely tied to accuracy (Tian et al., 2023). Or one can read the probability of each allowed answer at the answer position, which needs no generation but inherits the model's quirks: post-trained models are over-confident (OpenAI, 2023; Xiao et al., 2025; Leng et al., 2025), favour particular option labels and positions (Zheng et al., 2024; Pezeshkpour & Hruschka, 2024), and assign probabilities that contradict each other (Zhu & Griffiths, 2024; Andrews & Sarkar, 2026). These failures have mostly been studied one at a time and on fully post-trained models, which cannot say which step of post-training causes them, and so where a fix belongs. Three open pipelines now

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.

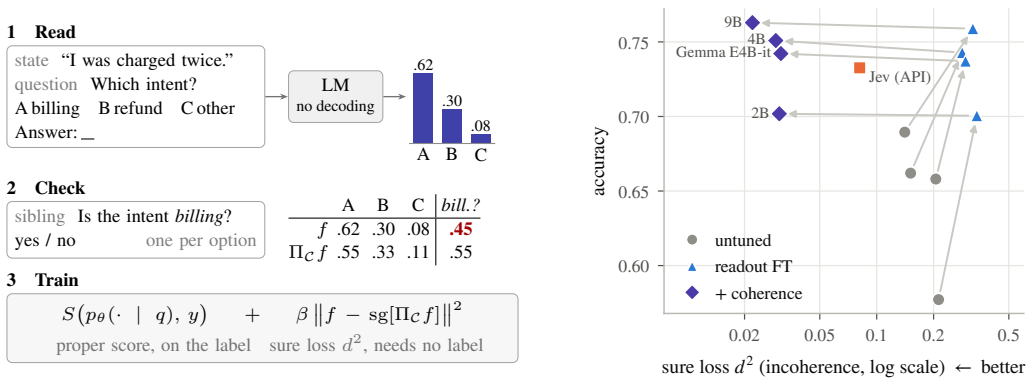


Figure 1: **Readout fine-tuning buys accuracy at the price of coherence, unless coherence is asked for.** *Left:* a decision is the distribution over the allowed answers at the answer cue, read without generating; the answers to a question and its siblings form a vector f, whose projection $\Pi_C f$ onto the coherent set is the target of a penalty added to a proper scoring rule. *Right:* each path runs from a model's untuned readout (gray) through fine-tuning with the proper scoring rule (blue) to the same with the penalty (violet), which removes the incoherence fine-tuning adds at unchanged accuracy, well below the commercial API (orange).

publish every intermediate checkpoint (Lambert et al., 2024; Team OLMo et al., 2024; Hugging Face, 2025), so the step can be found. We ask two questions: what does each stage of post-training do to a model's *decisions*, as opposed to its generations, and can a model then be post-trained for decisions directly?

We answer both with one measurement. Every model is read the same way, from the probability of each allowed answer at the answer position, with a small recipe fitted on validation data only (Section 2), and scored on DECIDE-22, a benchmark we build from 22 public tasks with 22,773 test items, four with human vote distributions and six held out of all fine-tuning (Section 3). Besides accuracy and calibration we measure invariance to how the options are presented, and coherence: whether a model's answers to a question and to the questions it implies could come from a single belief, quantified by de Finetti's sure loss (de Finetti, 1974).

Tracing 26 checkpoints from six open families through every stage they release (Section 4), we find that base models disagree, some under- and some over-confident, but the step after supervised fine-tuning does not. In all three pipelines that publish their intermediate checkpoints, preference optimization (DPO in Tülu 3 and OLMo 2, APO in SmolLM3; Lambert et al. 2024; D'Oosterlinck et al. 2025) sharpens every question type at unchanged accuracy, raising the fitted temperatures 1.3- to 2.4-fold, and the change is almost exactly a temperature: one rescaling of the supervised model's log-odds explains over 90% of their variance afterwards, against 45–70% for supervised fine-tuning itself.

We then post-train for decisions directly (Section 5). Readout fine-tuning trains the model on its own answer distribution with a proper scoring rule (Gneiting & Raftery, 2007). It adds 7 to 22 points of accuracy, brings calibration error to 0.05–0.07, and undoes the preference step: started from a preference-optimized checkpoint or from its SFT parent, it reaches the same accuracy and the same temperatures. But it also makes the model contradict itself more than before (Figure 1). We therefore add a penalty on the sure loss of each training question's automatically derived siblings. It needs no extra labels, and its target is safe by construction: projecting incoherent answers onto the coherent set lowers the Brier score whatever the truth. The penalty cuts incoherence tenfold at no cost in accuracy, in four model families from 2B to 9B parameters. A Qwen3.5-9B model trained this way reaches 76.0% accuracy and a calibration error of 0.054 on DECIDE-22 (means over three seeds), against 73.3% and 0.113 for Jev as served (0.068 with our recipe), with distributions closer to human votes. Jev keeps its lead on the held-out tasks, mostly from one rating scale, and on applying a rule stated in the question.

Contributions.

- **A benchmark for decisions, and the case against decoding.** We build DECIDE-22: 22 typed decision tasks (22,773 test items, six held out from all fine-tuning, four with human vote distributions) and 4,749 automatically generated families of logically related questions, scored for accuracy, calibration, agreement with people, invariance and coherence. On it, decoding an answer costs several times more than reading it and yields a confidence that barely separates right from wrong answers, before or after reasoning (Sections 2, 3 and 5.1).
- **Where post-training bends decisions.** Over-confidence after alignment is known; across 26 checkpoints and three pipelines we locate it in the preference step, show that it acts on the readout almost exactly as a temperature (R^2 0.92–0.95), and show that readout fine-tuning erases it (Sections 4 and 5).
- **Readout fine-tuning with a coherence penalty.** Coherence as a label-free regularizer has been proposed in theory (Andrews, 2026). We make it a training objective with an exact gradient and a target that provably improves the Brier score, show that ordinary decision fine-tuning is what makes readouts incoherent, and remove that incoherence tenfold without costing accuracy. A 9B model trained this way beats a commercial decision API on accuracy and calibration (76.0% and 0.054 against 73.3% and 0.113, or 0.068 when the API gets our recipe) with under a third of its incoherence; on the same 4B backbone it beats a concurrent open decision model trained on six times the data (Section 5).

2 READING A DECISION

A *decision* here is a typed question about a piece of text, the *state*, with a closed set of allowed answers. We use three types, following Jev: a *choice* among K named options (an intent, a label, an exam answer), a *score* on K ordered levels (a rating, a severity), and a yes/no question whose output is a single probability. The model must return a probability distribution over the allowed answers.

The readout. We render the state, the question and the allowed answers into a fixed prompt that ends with an answer cue, wrap it in the model's chat template, and run one forward pass. The options of a choice are listed under identifiers that are single tokens for the model's tokenizer (A–Z, then two-letter codes, so that K up to 151 still fits one step); levels are read as digits; yes/no as the words *yes* and *no*. The decision is the softmax of the answer tokens' logits at the cue, restricted to the allowed answers:

$$p_\theta(a_k \mid s, q) = \frac{\exp(\ell_k/T)}{\sum_{j=1}^{K} \exp(\ell_j/T)}, \qquad \ell_k = \log \pi_\theta\big(t(a_k) \mid \mathrm{prompt}(s, q)\big), \tag{1}$$

where $t(a_k)$ is the token of answer a_k. Nothing is generated and nothing is added to the network: the cost of a decision is the cost of reading the prompt.

The recipe. Raw readouts are biased in known ways (Zhao et al., 2021; Zheng et al., 2024), so every model gets a small recipe fitted on validation data only (up to 200 items per task; no test label enters it): for choice questions the probabilities are averaged over two presentations of the options in different orders (a second forward pass in the same batch; dropping it costs the fine-tuned 9B model 0.3 points of accuracy and nothing in calibration), a temperature T is fitted per question type by negative log-likelihood, and yes/no questions get a bias on the log-odds, because a temperature cannot move a decision threshold. The fitted temperature is also a diagnostic we use throughout: $T > 1$ means the raw readout was over-confident, $T < 1$ that it was under-confident. Nothing else about a model is tuned.

Why not decode? Reading costs what the prompt costs: a single-pass Qwen3.5-9B decision takes a median 63 ms on one A40, against 455 ms to decode it with a stated confidence and about two minutes after reasoning (Section 5.1).

3 THE BENCHMARK AND THE MEASUREMENT

Tasks. We build DECIDE-22 from 22 decision tasks in public datasets (Table 2): nine choice tasks (intent routing with up to 151 intents, legal provisions, emotions, inference, exam questions),

six score tasks (ratings, response quality, similarity, hate-speech severity) and seven yes/no tasks (question answering, fact verification, paraphrase, toxicity, spam), with 22,773 test items and their own training and validation splits. Six tasks are *held out from fine-tuning*: no fine-tuning in this paper trains on them, so they measure transfer to new tasks, new label sets and, in two cases, new ordinal scales. Like every task with a validation split, they contribute up to 200 validation items to fitting each model's recipe, whose only parameters are three temperatures and a yes/no bias shared across all tasks; nothing task-specific is fitted. Four tasks come with the vote distribution of their human annotators: ChaosNLI, which has 100 annotators per item (Nie et al., 2020), GoEmotions, Civil Comments and Measuring Hate Speech.

Accuracy, calibration and agreement with people. We report accuracy and the expected calibration error (Guo et al., 2017) of the top answer (15 equal-width bins), each computed per task and averaged over the 22 tasks, so that no single large task dominates. Where annotators disagree, a calibrated model should not be certain, and matching the majority label is the wrong target (Baan et al., 2022). On the four tasks with vote distributions we therefore also report the total variation distance (TVD) between the model's distribution and the annotators'.

Invariance. A decision should not depend on how its options are presented. We measure how often the top answer changes when the options are shown in a different order (*order flips*), how much accuracy survives when option names are replaced by opaque identifiers, how much is lost going from two options to a task's full set, and how the answer moves when the identifiers change style (letters, digits, symbols).

Coherence. One question implies others: "which intent?" implies "is it *billing*?"; "is the claim supported?" implies "is it *not* supported?"; "which level?" implies "is it at least level 3?". We build such families automatically from each test question: a choice is paired with each option asked as a yes/no question (all options when $K \leq 4$, otherwise the correct one and three others), a yes/no question with its negation and with the same statement as a true/false choice, and a score with up to four threshold questions. This gives 4,749 families and 19,595 questions from 150 test items per task. Every answer in a family is an *event*, a set of possible worlds $w \in \mathcal{W}$, where a world is the true answer of the original question. Collect the probabilities a model states for these events in a vector $f \in [0,1]^m$ and let $V \in \{0,1\}^{m \times |\mathcal{W}|}$ indicate which worlds make each event true. The answers are *coherent* if some belief over worlds explains all of them, that is if f lies in $\mathcal{C} = \{Vp : p \in \Delta(\mathcal{W})\}$. We measure incoherence by the *sure loss*

$$d^2(f) = \min_{p \in \Delta(\mathcal{W})} \|f - Vp\|_2^2. \tag{2}$$

The name is literal: de Finetti's theorem says that an incoherent f can be beaten, whatever the truth, by a coherent one (de Finetti, 1974). With the Brier score the projection $f^\star = \Pi_{\mathcal{C}}(f)$ does exactly that: because every world's truth vector Ve_w lies in the convex set $\mathcal{C}$,

$$\|f^\star - Ve_w\|^2 \;\leq\; \|f - Ve_w\|^2 - d^2(f) \qquad \text{for every world } w, \tag{3}$$

so a model with sure loss d^2 loses at least d^2 in Brier score against its own coherent correction, whichever answer turns out to be right. $d^2 = 0$ is coherent; the untuned checkpoints we measure range from 0.14 to 0.43.

Out of distribution, and the reference. Three further tests (Appendix F) probe applying a rule stated in the question, choosing "none of the above" when the correct option has been removed, and ignoring instructions injected into the state. We score Jev 1.13.0 on the full battery through its public API (TypeSafe, 2026); its readout is not observable, so it enters every comparison as a black box.

4 What post-training does to decisions

We score 26 checkpoints from six open families at every post-training stage they publish: Qwen3.5 base and instruct at 2B, 4B and 9B; Qwen3-4B base, instruct and thinking (Yang et al., 2025); Gemma-4 E2B and E4B, base and instruction-tuned; Llama-3.1-8B (Grattafiori et al., 2024) and the three stages of Tülu 3 built on it (SFT, DPO, RLVR; Lambert et al. 2024); OLMo 2 7B and the same three stages, which follow the Tülu 3 recipe (Team OLMo et al., 2024); and SmolLM3-3B

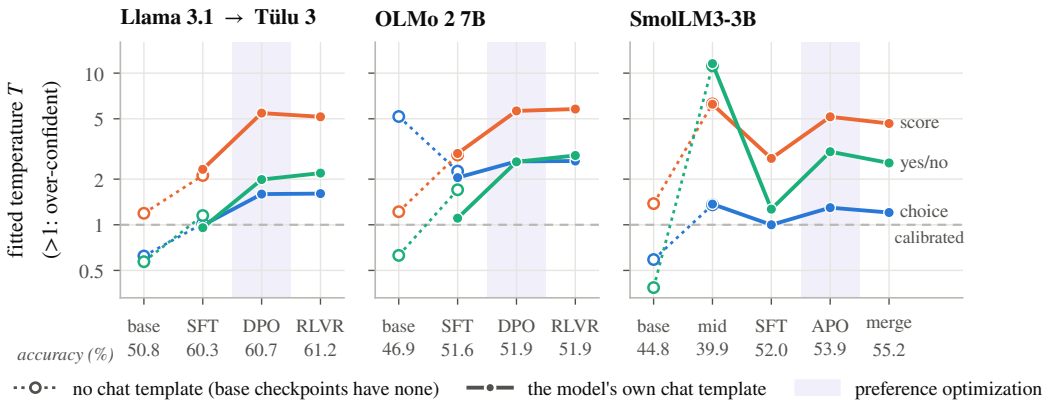


Figure 2: **What each post-training stage does to the decision readout**, in the three open pipelines that release their intermediate checkpoints: the temperature fitted on validation per question type ($T > 1$: the raw readout was over-confident), with each stage's accuracy under its name (uniform guessing: 27.5%). In all three, preference optimization (shaded) raises every temperature at flat accuracy, and RLVR leaves it where DPO put it.

at base, mid-training, SFT, preference optimization (APO, D'Oosterlinck et al. 2025) and the final merge (Hugging Face, 2025). Every checkpoint is read the same way, with its own recipe. Base checkpoints have no chat template, so where a step crosses from base to post-trained we read both without a template, and where the post-trained stages have one we read them in it (Appendix B).

Base models do not agree; what follows them does. Fitted choice temperatures of the base checkpoints run from 0.59 (SmolLM3, under-confident) to 5.2 (OLMo 2, over-confident), so there is no single "base model" calibration to preserve or destroy. What replicates is the step after them (Figure 2). Supervised fine-tuning leaves the choice readout close to calibrated in two of the three pipelines ($T = 0.99$ for Tülu, 1.00 for SmolLM3, 2.05 for OLMo 2), and score questions over-confident in all three ($T = 2.3$–3.0). Preference optimization then sharpens every question type at unchanged accuracy in all three, raising the fitted temperatures 1.3- to 2.4-fold for $+0.4$ and $+0.3$ points of accuracy with DPO and $+1.8$ with APO: two labs, three base models and two preference algorithms. RLVR after DPO leaves the temperature where DPO put it in both ladders that have it, and the most over-confident readouts in the study belong to instruction-tuned checkpoints (Gemma-4-E2B-it and Qwen3-4B-Instruct-2507, T up to 14 and 13 in their own templates; Table 3).

Preference optimization is almost a temperature. If a stage only rescaled the readout, the later checkpoint's centred log-scores would be a multiple of the earlier one's, and the points in Figure 3 would lie on lines through the origin. Over all 22,773 test items this holds with $R^2 = 0.92$ for Tülu's DPO step, 0.95 for OLMo 2's and 0.93 for SmolLM3's APO step, with slopes of 1.1 to 2.4 by question type: the answers keep their order and become sharper, which is why accuracy barely moves. Supervised fine-tuning is a different kind of change (R^2 0.45–0.70): it reorders answers. RLVR after DPO barely moves the readout (R^2 0.98 and 0.99, slope 1.0). "Almost" matters: 12–18% of top answers still change, and the best single temperature removes only a third to a half of the probability change. Invariance and the chat template are in Appendix C. Preference optimization's effect on coherence does not replicate (sure loss $+0.07$ and $+0.02$ after DPO, -0.05 after APO), and we make no claim about it.

5 Post-training for decisions

If the readout is the product, it can be trained directly. We keep the architecture and the prompt of Section 2 and fine-tune the model on its own decision distribution.

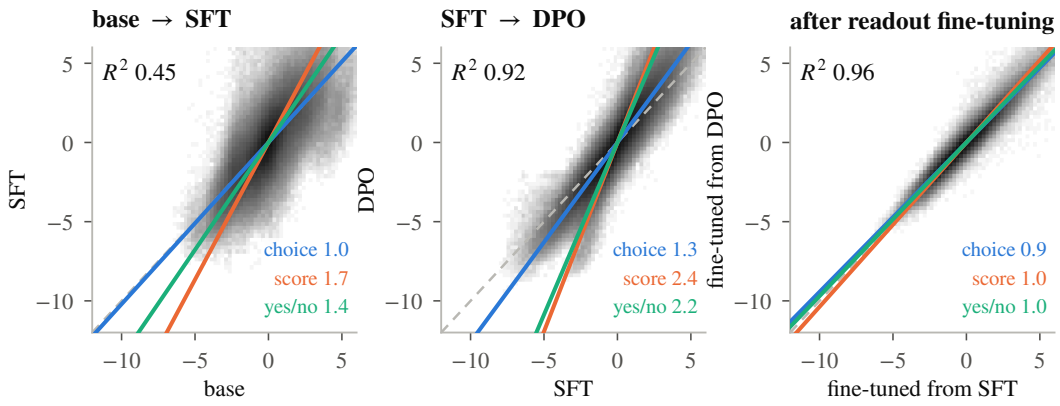


Figure 3: **Preference optimization is almost a temperature, and readout fine-tuning erases it** (Tülu 3; OLMo 2 and SmolLM3 in Appendix C). Centred log-scores of each checkpoint against the one before it, over every test item and allowed answer (gray: density, log scale), with least-squares lines through the origin per question type (slope = sharpening; dashed: no change). SFT reorders answers; DPO keeps their order and sharpens them; after readout fine-tuning, the models started from SFT and from DPO agree at slope 1.

Readout fine-tuning. For a labelled question the loss is a strictly proper scoring rule (Gneiting & Raftery, 2007) applied to the restricted distribution of Equation 1 at $T = 1$: the log score for choice and yes/no questions and the ranked probability score for scores, which respects the order of the levels. The options are shuffled for every training example, so that position cannot become a feature. We train on the training splits of the 16 non-held-out tasks, at most 400 questions per task (5,885 in total), with a rank-16 LoRA adapter on every projection (Hu et al., 2022), learning rate 3×10^{-5}, two epochs, and the epoch chosen by validation loss. A full fine-tuning variant trains every weight (fp32 master weights, learning rate chosen on validation). The fine-tuned model is read exactly as in Section 2, with a recipe refitted on validation.

The coherence penalty. Supervised training scores each question alone, and nothing asks the model to keep its answers to related questions consistent. We therefore train on each training question's whole family and add its sure loss to the objective,

$$\mathcal{L}(\theta) = \sum_{q} S\big(p_\theta(\cdot \mid q), y_q\big) \;+\; \beta \sum_{F} \big\| f_F(\theta) - \mathrm{sg}\big[\Pi_{\mathcal{C}_F}(f_F(\theta))\big] \big\|_2^2, \tag{4}$$

where $f_F(\theta)$ stacks the probabilities the model assigns to the events of family F, $\Pi_{\mathcal{C}_F}$ is the projection onto its coherent set (a small quadratic program), and sg stops the gradient. Because $\mathcal{C}_F$ is convex, the gradient of the penalty, $2\,(f_F - \Pi_{\mathcal{C}_F} f_F)$, is exactly the gradient of the sure loss $d^2(f_F)$, not a surrogate. The penalty reads no labels beyond the original question's, which the supervised term already uses, and by Equation 3 its target is never worse than the model's current answers, whatever the truth; the guarantee is for the target, not the trained model. We use $\beta = 1$.

Fine-tuning repairs calibration and invariance, but costs coherence. From every starting point readout fine-tuning adds 7 to 22 points of accuracy (the most for the checkpoints that start lowest), brings calibration error to 0.05–0.07, cuts the order flips (Qwen3.5-4B 0.14 → 0.07, 9B 0.12 → 0.05), and cuts the rate at which injected instructions hijack the answer roughly two- to fivefold (Figure 4). But the supervised model contradicts itself: its sure loss ends at 0.27–0.37, above the untuned model's from every post-trained start. Training each question alone sharpens its answer without keeping the answers to its siblings in line.

The penalty removes that at no cost in accuracy. With the penalty the sure loss falls about tenfold, to 0.02–0.04, below Jev's 0.081 overall and on each of the three family types, in every model family and under both LoRA and full fine-tuning. This is not the penalty grading its own homework: the sure loss is measured on families built from test items, including all six held-out

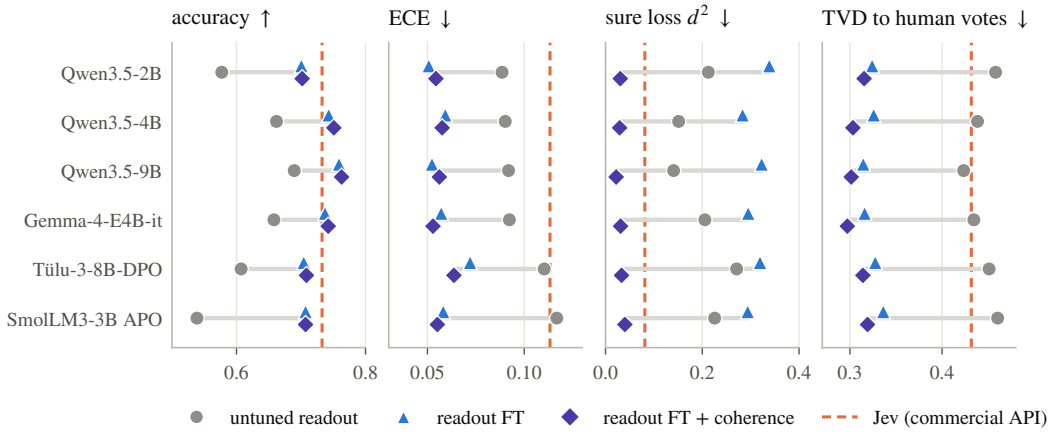


Figure 4: **Readout fine-tuning with and without the coherence penalty.** One row per starting model (the SFT starts of Tülu and SmolLM3 are in Table 4); the dashed line is Jev. Fine-tuning adds accuracy and brings calibration error to 0.05–0.07 in every family, but raises the sure loss; the penalty removes it at the same accuracy.

Table 1: **Qwen3.5-9B against the commercial API** on the full DECIDE-22 test set. Fine-tuned rows are means ± s.d. over seeds 0, 1 and 2. *Jev with the open models' recipe (per-type temperature, yes/no bias), fitted on the same validation items. Last rows: first coherence-penalty run minus Jev as served, with 95% paired-bootstrap intervals (items resampled within tasks). Held-out: the six tasks held out from fine-tuning. TVD: distance to human votes (four tasks).

	Accuracy (%)	Held-out (%)	ECE	TVD	Sure loss
Jev 1.13.0 (API), as served	73.3	**83.5**	0.113	0.432	0.081
with our recipe*	73.2	83.1	0.068	–	–
Qwen3.5-9B, untuned	68.9	74.5	0.092	0.423	0.141
+ readout fine-tuning	75.9 ±0.4	79.6 ±1.2	0.056 ±.006	0.327 ±.014	0.291 ±.032
+ coherence penalty	**76.0** ±0.5	79.9 ±0.5	**0.054** ±.002	**0.319** ±.016	**0.024** ±.002
difference to Jev	+3.0	−3.2	−0.057	−0.130	−0.059
95% interval	[2.5, 3.6]	[−4.1, −2.2]	[−.066, −.054]	[−.137, −.123]	[−.062, −.056]

tasks, none of which the penalty saw in training. Nor does it rest on the families whose siblings are chosen with the label (choices with more than four options): on the sixteen tasks whose families are built without it, the 9B model's sure loss falls from 0.34 to 0.016 (Appendix E). Accuracy is unchanged within seed variation (2B, three seeds: 69.5 ± 0.6% supervised against 70.1 ± 0.8% with the penalty; 4B, two seeds: 74.8 ± 0.5% against 74.8 ± 0.3%; 9B, three seeds: 75.9 ± 0.4% against 76.0 ± 0.5%). Full fine-tuning lands where LoRA does (4B: 75.2%, and 75.7% with the penalty), and so does Qwen3.5-4B-Base, although the instruct checkpoint started 1.5 points ahead of it (Table 4). Table 1 compares the largest model with Jev.

Where the API still leads. On the six held-out tasks Jev leads the fine-tuned 9B model by 3.7 points, 3.2 of them from one task (Table 9). On Yelp star ratings, readout fine-tuning *lowers* accuracy (53.7% → 49.5%, against 68.5% for Jev) but not the ordering (Spearman 0.88–0.89 against 0.87; Jev 0.93): fine-tuning on other scales moved this one's level boundaries, and one bias per level fitted on 16 labelled reviews brings it back to 53% (untuned: 53.7%; Appendix F). On the other five the model trails Jev by 0.6 points on average, behind on four and ahead on hate-speech severity (59.1% against 52.7%). Given the same recipe, Jev is also better calibrated on the held-out tasks (0.061 against 0.070); our lead comes from the sixteen fine-tuning tasks (0.048 against 0.071).

Against an open decision model. Tev1-4B (Together AI, 2026), a concurrent open decision model, fine-tunes the same Qwen3.5-4B to emit the answer letter, on 37,840 questions: six times

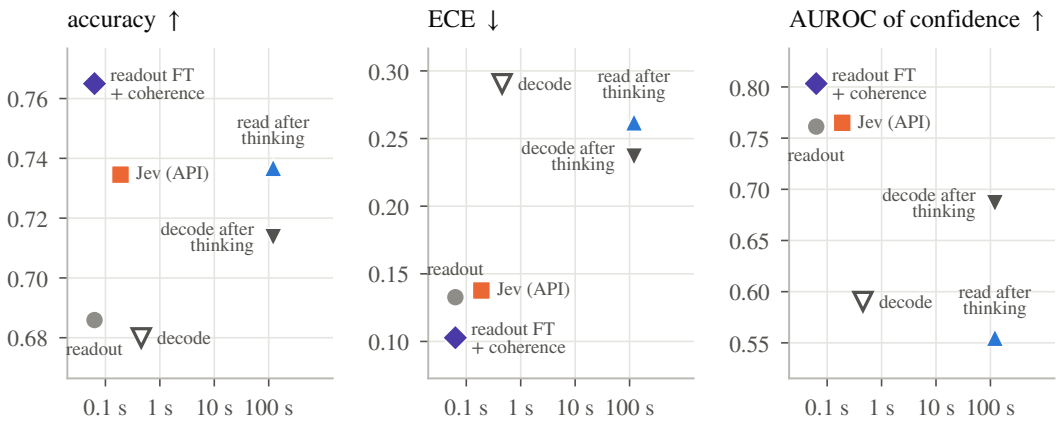


Figure 5: **Decide, don't decode.** Qwen3.5-9B answers the same 2,200 test items by reading the allowed answers (circle), by decoding its answer with a stated confidence (hollow triangle), by decoding after reasoning (filled triangle), or by reading the answer distribution after the same reasoning (blue); the fine-tuned readout (diamond) and Jev (square) for reference. Time: median, batch size one, one A40; for Jev, a network round trip to its API.

our data, including the training splits of four of our tasks. Read in its own prompt format with our recipe, it reaches 70.3% with calibration error 0.086 and sure loss 0.239, and 76.6% on the held-out tasks. Readout fine-tuning with the penalty reaches 75.1%, 0.058, 0.029 and 79.2% (Table 4).

Readout fine-tuning undoes what preference optimization did. Fine-tuning the readout from the preference-optimized checkpoint and from its SFT parent, with the same recipe and seed, lands both in the same place: 70.4% and 70.8% in Tülu 3, 70.7% and 70.5% in SmolLM3, 70.3% and 70.5% in OLMo 2. The two fine-tuned readouts agree with R^2 of 0.96–0.98 and slopes of 0.94–1.11 in all three ladders, against 1.1–2.4 before (Figure 3, right): the preference step leaves almost no trace in the decision. With the penalty the two SmolLM3 starts again end level (70.7% and 71.0%); in Tülu the SFT start ends 1.4 points ahead (72.2% against 70.8%; seed standard deviations are 0.3–0.8 points).

What the penalty does not do. The penalty is not a general coherence prior. Holding one family type out of it (the model still trains on that type's supervised loss) leaves that type largely incoherent: it gets -14%, 9% and 21% of the full penalty's reduction for score, yes/no and choice families at 2B, and -28%, 10% and 44% at 4B (two runs each; choice families contain yes/no members the other penalty trains), while the penalized types become coherent and accuracy does not move (Appendix E). It generalizes to new questions and tasks, not to a new *kind* of logical relation; relations have to be added to the family generator. Nor is coherence calibration: the penalty on unlabelled families from the held-out tasks lowers the sure loss further (0.021 against 0.033) but more than doubles their calibration error (0.055 → 0.133), so we penalize only labelled training questions (weight ablation in Appendix E).

5.1 Deciding against decoding

Is reading the answer a compromise forced by latency, or is it simply better? We asked Qwen3.5-9B the same 2,200 test items in four ways (Figure 5). Decoding the answer with a stated confidence (Tian et al., 2023) is exactly as accurate as the readout (68.0% against 68.6%), but the stated confidence is 100 on most items and 90–95 on nearly all the rest, so it barely separates right answers from wrong ones (AUROC 0.59 against 0.76). Reasoning first buys three points, mostly on knowledge questions, at about two thousand times the cost, and its stated confidence is still uninformative (AUROC 0.69). Reading the answer distribution *after* the reasoning is the most accurate way to use

it (73.7%, and level with Jev on the held-out tasks), but the distribution has collapsed: 95% of items receive a probability of at least 0.999, the margins no longer rank errors (AUROC 0.56), and no temperature fitted on the other half of the items recovers calibration. Reasoning does not only cost time; it consumes the uncertainty that a single forward pass still carries. The fine-tuned readout is the most accurate of all (76.5%) at the cost of one or two forward passes. The pattern repeats at 4B (Appendix G).

6 Related work

Calibration of language models. Pretrained models are often well calibrated on multiple-choice questions (Kadavath et al., 2022), and alignment degrades this (OpenAI, 2023; Zhu et al., 2023); recent work repairs the over-confidence during RLHF or with calibration-aware fine-tuning (Leng et al., 2025; Xiao et al., 2025; Kapoor et al., 2024). We locate the inflation in the preference step and undo it by fine-tuning the readout. Verbalized confidence (Tian et al., 2023; Xiong et al., 2024) is the alternative for black-box models; Section 5.1 measures what it loses for open ones.

Multiple-choice readouts. Scoring options by their identifiers is standard in multiple-choice evaluation (Robinson & Wingate, 2023), and its biases and need for calibration are well documented (Zheng et al., 2024; Pezeshkpour & Hruschka, 2024; Zhao et al., 2021). We treat the readout as the product rather than an evaluation device, and train it directly, as decision models sold as a service (TypeSafe, 2026) or released openly (Together AI, 2026) do; we compare with both. Appendix H covers further work.

Coherence and consistency. Language models contradict themselves across paraphrases and implications (Elazar et al., 2021; Kassner et al., 2021), assign probabilities that violate the laws of probability (Zhu & Griffiths, 2024; Fluri et al., 2024), and can be exploited by Dutch books (Andrews & Sarkar, 2026). Consistency has been trained with losses derived from logical rules (Li et al., 2019; Asai & Hajishirzi, 2020), a paraphrase-consistency loss (Elazar et al., 2021) and a semantic loss over given constraints (Calanzone et al., 2025), and restored at inference by constraint solving over a model's answers (Kassner et al., 2021; Mitchell et al., 2022) or by projecting stated probabilities onto the coherent set (Chadwick et al., 2025). The closest trained loss is label-free: contrast-consistent search trains a probe so that a statement and its negation get complementary probabilities (Burns et al., 2023), and for one such pair its consistency term is exactly twice our sure loss. Most directly, Andrews (2026) proposes de Finetti coherence as a label-free regularizer for language models, in theory only. We make it a training objective for the model's own readout: any finite family of events, with the projection as the target, which gives the exact gradient of the sure loss and a target that improves the Brier score in every world (Equation 3), families generated from each question's type, and a proper scoring rule alongside. Unlike an inference-time repair, the penalty needs no sibling answered at inference.

7 Limitations and conclusion

Our fine-tuned models are trained on 16 task distributions. The six held-out tasks show that the gains transfer, with one clear exception: a rating scale that resembles the trained ones but is used differently, where fine-tuning hurts; on these tasks Jev given our recipe is also better calibrated. Jev also leads on applying a stated rule. The penalty covers only the relations our family generator builds (Section 5). Fine-tuning also moves the decision threshold of an unseen phishing benchmark (AreLit, 2026; anisselbd, 2026) while keeping its ranking; one log-odds shift fitted on 16 labelled emails repairs it (Appendix F). The post-training analysis rests on three pipelines with published intermediate checkpoints, two of them from one lab. Finally, a readout that keeps its uncertainty through reasoning is the natural next step.

An open model's readout can be measured as a decision model, its calibration is set by identifiable stages of post-training, and a short, targeted post-training step makes it more accurate and better calibrated overall, and far more coherent, than a commercial decision API, without generating a single token.

Reproducibility statement

All tasks in DECIDE-22 are built from public datasets with fixed seeds (Appendix A); every checkpoint is named, with commits for the intermediate stages (Appendix B); recipes are fitted on validation data only; and every number in the paper is computed from stored per-item predictions by scripts we release with the benchmark and the trained models. Fine-tuning runs take one to six hours on a single A40 or L40S GPU (the 4B full fine-tune uses two). An anonymized copy of the code is in the supplementary material and at https://anonymous.4open.science/r/decide22-code-CED2.

Ethics statement

The tasks include toxicity, hate speech and spam detection, drawn from existing public datasets under their licenses. A calibrated decision model makes it easier to automate such judgements; we see calibration and coherence as prerequisites for handing a decision to a model responsibly, not as a reason to remove people from it.

AI use statement

In this work, we used generative AI tools (an LLM-based coding agent, working under the authors' direction) to implement and run the experiment, training and analysis code, to analyse results and produce the figures, to propose analyses and controls, which the authors selected, and to draft and edit the text. We did not use generative AI to create or label data: the benchmark is built from public datasets, and the language models evaluated in the paper are research subjects, not tools. We have reviewed all AI-assisted work: every number in the paper was traced to stored per-item predictions and result files, the analyses behind the main claims were rerun as controls, and the references were checked against their sources. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

References

Tiago A. Almeida, José María Gómez Hidalgo, and Akebo Yamakami. Contributions to the study of SMS spam filtering: New collection and results. In *Proceedings of the ACM Symposium on Document Engineering (DocEng)*, 2011.

Isaiah Andrews. Revealed rationality: Label-free evaluation and regularization from representation theorems. *arXiv preprint arXiv:2608.05015*, 2026.

Isaiah Andrews and Suproteem Sarkar. Dutch books for language models. *arXiv preprint arXiv:2609.02797*, 2026.

anisselbd. jev-phishing-bench: Jev (TypeSafe) vs Claude Haiku 4.5 on 2 000 phishing emails. https://github.com/anisselbd/jev-phishing-bench, 2026. Evaluation repository built on the PhishNChips dataset; commit 1d56e8c.

AreLit. PhishNChips: a phishing email dataset. https://huggingface.co/datasets/AreLit/PhishNChips, 2026. Hugging Face dataset, version 5.2.

Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernández. Stop measuring calibration when humans disagree. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of the World Wide Web Conference (WWW)*, 2019.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

Diego Calanzone, Stefano Teso, and Antonio Vergari. Logically consistent language models via neuro-symbolic integration. In *International Conference on Learning Representations (ICLR)*, 2025.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, 2017.

Alina Chadwick, Anson Kahng, and Jens Kipper. Dutch books and money pumps: Rectifying vulnerabilities in LLMs through rationality. In *Proceedings of the Conference on Human and Artificial Rationality (HAR)*. Springer, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Bruno de Finetti. *Theory of Probability: A Critical Introductory Treatment*. Wiley, 1974.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *Transactions of the Association for Computational Linguistics*, 2025.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

Jack FitzGerald et al. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Lukas Fluri, Daniel Paleka, and Florian Tramèr. Evaluating superhuman models with consistency checks. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2024.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Neel Guha et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Hugging Face. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Chris J. Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. Constructing interval variables via faceted Rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*, 2020.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *International Conference on Learning Representations (ICLR)*, 2025.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D. Manning. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Yixin Nie, Xiang Zhou, and Mohit Bansal. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL*, 2024.

Joshua Robinson and David Wingate. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations (ICLR)*, 2023.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Together AI. Tev1: open-weight, Jev-inspired decision model finetuned on top of Qwen3.5 4B. https://github.com/togethercomputer/tev1, 2026. Training code; weights at https://huggingface.co/togethercomputer/Tev1-4B-experimental, revision 0b7becf; released September 2026.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2020.

TypeSafe. Jev: System One decision models. https://typesafe.ai, 2026. Commercial API, version 1.13.0; accessed September 2026.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Jiancong Xiao, Bojian Hou, Zhanliang Wang, Ruochen Jin, Qi Long, Weijie J. Su, and Li Shen. Restoring calibration for aligned large language models: A calibration-aware fine-tuning approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*, 2024.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Jian-Qiao Zhu and Thomas L. Griffiths. Incoherent probability judgments in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2024.

A THE TASKS

Table 2 lists the 22 tasks. Each task keeps its source dataset's label set; training, validation and test items are sampled with a fixed seed from the source's own splits (a task whose source has no validation split takes it from the training split). ChaosNLI has only a test split, so it contributes nothing to any recipe. The six held-out tasks are never used for fine-tuning or for the coherence penalty, and their validation items are used only to fit each model's shared recipe. The coherence families (Section 3) are built from 150 test items per task and all ChaosNLI items.

Sources. Banking77 (Casanueva et al., 2020), CLINC150 (Larson et al., 2019), MASSIVE (FitzGerald et al., 2023), LEDGAR (Tuggener et al., 2020), GoEmotions (Demszky et al., 2020), MMLU (Hendrycks et al., 2021), ARC-Challenge (Clark et al., 2018), MNLI (Williams et al., 2018), ChaosNLI (Nie et al., 2020), SST-5 (Socher et al., 2013), Yelp reviews (Zhang et al., 2015), HelpSteer2 (Wang et al., 2024), STS-B (Cer et al., 2017), Measuring Hate Speech (Kennedy et al., 2020), BoolQ (Clark et al., 2019), FEVER (Thorne et al., 2018), PAWS (Zhang et al., 2019), Civil Comments (Borkan et al., 2019), SMS Spam (Almeida et al., 2011) and StrategyQA (Geva et al., 2021), each loaded from its public release on the Hugging Face Hub; the builder in the code release names each one.

Label conversions. Most tasks keep their source labels. Four are converted. *GoEmotions* is rebuilt from the raw per-rater annotations: comments with fewer than three raters, or marked "very unclear" by any rater, are dropped; the vote distribution is each emotion's share of all rater selections (a rater may select several), and the label is the emotion with the most selections, ties going to the emotion that is more frequent overall. *Measuring Hate Speech* aggregates annotators' three-level ratings per comment (at least three annotators): the vote distribution is the share of ratings at each level, the label its most frequent level (ties to the middle level), and the comments are split 85/5/10 into train, validation and test by a seeded shuffle, since the release has one split. *STS-B* rounds the 0–5 similarity score to six levels. *Civil Comments*' vote distribution is the share of annotators who marked the comment toxic. FEVER keeps only supported and refuted claims.

Table 2: DECIDE-22 tasks. K: number of allowed answers (ARC-Challenge varies by question). Votes: the item comes with its annotators' vote distribution. Held out: never fine-tuned on (their validation items still fit the shared recipe).

Task	Type	K	Train	Validation	Test	Votes	Held out
Banking77	choice	77	8000	500	1000		
CLINC150	choice	151	8000	500	1000		✓
MASSIVE (intent)	choice	60	8000	500	1000		
LEDGAR	choice	100	8000	500	1000		
GoEmotions	choice	28	8000	500	1000	✓	
MMLU	choice	4	285	500	1000		
ARC-Challenge	choice	3–5	1119	299	1000		✓
MNLI	choice	3	8000	500	1000		
ChaosNLI	choice	3	–	–	1599	✓	
SST-5	score	5	8000	500	1000		
Yelp reviews	score	5	8000	500	1000		✓
HelpSteer2 helpfulness	score	5	8000	500	1000		
HelpSteer2 verbosity	score	5	8000	500	1000		
STS-B	score	6	5749	500	1000		
Measuring Hate Speech	score	3	8000	500	1000	✓	✓
BoolQ	yes/no	2	8000	472	1000		
FEVER (gold evidence)	yes/no	2	8000	500	1000		✓
PAWS	yes/no	2	8000	500	1000		
Civil Comments	yes/no	2	8000	500	2000	✓	
SMS Spam	yes/no	2	4000	237	800		
StrategyQA (closed book)	yes/no	2	1400	160	687		
StrategyQA (with facts)	yes/no	2	1400	160	687		✓
Total					22,773	4	6

B Models and prompts

Every checkpoint is loaded from the Hugging Face Hub (September 2026): Qwen3.5 2B, 4B and 9B, base and instruct; Qwen3-4B Base, Instruct-2507 and Thinking-2507; Gemma-4 E2B and E4B, base and instruction-tuned; Llama-3.1-8B (from a mirror whose weights are byte-identical to the gated release) and Tülu-3-8B SFT, DPO and RLVR; OLMo-2-1124-7B and its SFT, DPO and Instruct (RLVR) checkpoints; and SmolLM3-3B base, the mid-training, SFT and APO checkpoints of its training run, and the final merged model. Intermediate stages, several of which share one repository, are pinned to commits: SmolLM3 base `d78a42f`, mid-training `0485ec1`, SFT `f6ddaa5`, APO `cfb32d5` and final `a07cc9a`; OLMo 2 base `7df9a82`, SFT `1de02c0`, DPO `e34ea60` and Instruct `470b1fb`; and Tev1-4B `0b7becf`. The other checkpoints, one per repository, were loaded at their latest revision in September 2026. The Qwen3.5 base checkpoints ship their instruct model's chat template, and we read both base and instruct models in it. Llama-3.1-8B, OLMo 2 7B, the Gemma-4 bases and SmolLM3-3B base have no template; where a comparison crosses from them to a post-trained stage, both sides are read without one, and the post-trained stages are also read in their own template. The prompt is the same for every model: the state, the question, the options with their identifiers, the list of allowed answers and the cue "Answer:", under a one-sentence system instruction. Thinking is disabled in every chat template that supports it except where Section 5.1 enables it.

C The post-training trace in full

Table 3 gives every checkpoint of Section 4, and Figure 8 the temperature analysis behind its second paragraph: for consecutive checkpoints, the later one's centred log-scores regressed on the earlier one's through the origin, over all test items and their options. Figures 6 and 7 repeat Figure 3 for SmolLM3 and OLMo 2.

Invariance, and the chat template. The first post-training step makes decisions markedly more invariant: order flips fall in all eight base-to-post-trained pairs, often by half or more (Qwen3.5-2B 0.40 → 0.27; Llama-3.1-8B 0.50 → 0.31 after Tülu's SFT; Gemma-4-E4B 0.46 → 0.10), and the sure loss falls in seven of the eight. Some instruction-tuned models, however, only work inside their chat template: read without it, Gemma-4's instruct checkpoints are *worse* than their own bases (E2B 35.5% against 39.6%; E4B 31.7% against 48.8%), and in it they reach 59.1% and 65.8%. Comparing base and instruct models under one prompt is fair only for families that tolerate it. Preference optimization's effect on choosing "none of the above" also differs between the two ladders.

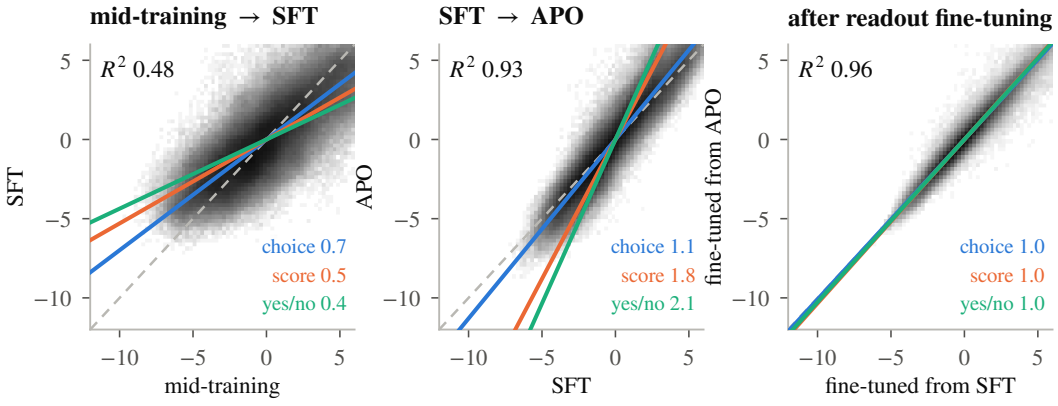


Figure 6: **The same for SmolLM3.** As Figure 3: the supervised step reorders answers, APO sharpens them in order, and readout fine-tuning from the SFT and from the APO checkpoint lands both on slope 1. Both sides of every panel are read in the model's chat template.

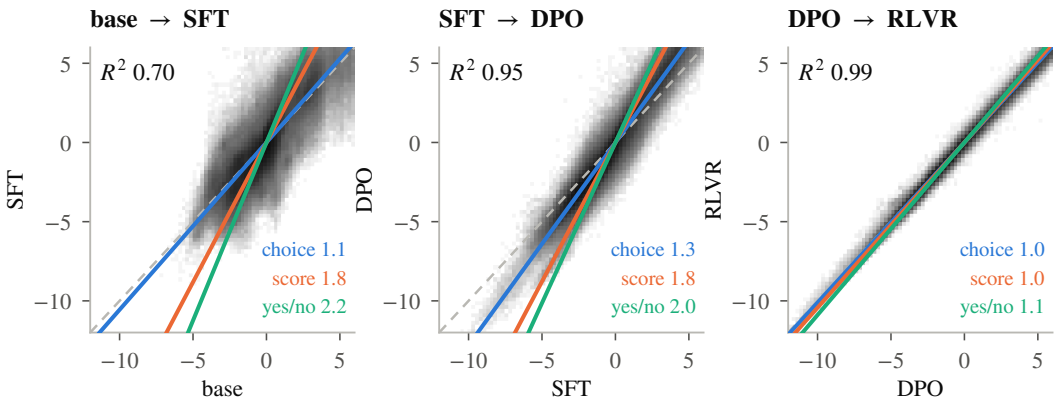


Figure 7: **The same for OLMo 2 7B**, whose post-training follows the Tülu 3 recipe: its supervised step is closer to a rescaling than in the other two ladders (R^2 0.70, read without a template on both sides), DPO sharpens the answers in order (R^2 0.95), and RLVR leaves them where DPO put them (R^2 0.99).

Table 3: Every checkpoint, read with its own recipe. T: fitted temperatures for choice / score / yes-no questions ($T > 1$: the raw readout was over-confident). Sure loss and order flips were not measured for the SmolLM3 rows read without a template. Guessing uniformly scores 27.5%. OLMo 2 and SmolLM3 lose most of their accuracy on the four intent and provision tasks with 60 to 151 options (16–41% after SFT, against 50–65% for Tülu 3); readout fine-tuning brings OLMo 2 to 73–79% on them.

Checkpoint	Acc. (%)	Held-out (%)	ECE	T	Sure loss	Order flip
Qwen3.5-2B-Base	53.3	57.6	0.124	1.73 / 1.95 / 0.70	0.377	0.402
Qwen3.5-2B	57.7	63.4	0.089	0.97 / 1.57 / 0.80	0.212	0.271
Qwen3.5-4B-Base	64.7	71.0	0.095	1.18 / 2.05 / 1.36	0.188	0.191
Qwen3.5-4B	66.2	71.9	0.090	1.24 / 2.16 / 1.14	0.151	0.140
Qwen3.5-9B-Base	67.9	72.5	0.096	1.05 / 1.69 / 0.97	0.169	0.147
Qwen3.5-9B	68.9	74.5	0.092	1.10 / 1.36 / 0.84	0.141	0.118
Qwen3-4B-Base	63.1	68.7	0.106	1.02 / 1.68 / 0.85	0.283	0.301
Qwen3-4B-Instruct-2507	64.6	71.7	0.108	4.71 / 12.99 / 12.58	0.224	0.173
Qwen3-4B-Thinking-2507	64.6	69.5	0.114	1.78 / 4.11 / 2.93	0.159	0.193
Gemma-4-E2B	39.6	38.3	0.115	0.70 / 0.80 / 1.41	0.161	0.613
Gemma-4-E2B-it, no template	35.5	35.4	0.089	7.13 / 6.15 / 20.00	0.190	0.693
Gemma-4-E2B-it	59.1	66.1	0.123	3.29 / 6.52 / 13.96	0.202	0.202
Gemma-4-E4B	48.8	55.4	0.155	1.26 / 0.89 / 0.36	0.435	0.457
Gemma-4-E4B-it, no template	31.7	40.7	0.063	20.00 / 6.53 / 20.00	0.171	0.767
Gemma-4-E4B-it	65.8	74.2	0.092	2.64 / 5.75 / 5.83	0.205	0.104
Llama-3.1-8B, no template	50.8	54.7	0.128	0.62 / 1.19 / 0.57	0.265	0.499
Tülu-3-8B-SFT, no template	60.5	64.7	0.111	1.02 / 2.11 / 1.15	0.173	0.314
Tülu-3-8B-SFT	60.3	64.4	0.099	0.99 / 2.32 / 0.96	0.200	0.311
Tülu-3-8B-DPO	60.7	63.7	0.110	1.59 / 5.45 / 1.99	0.271	0.308
Tülu-3-8B (RLVR)	61.2	64.5	0.108	1.61 / 5.15 / 2.19	0.214	0.299
OLMo-2-7B, no template	46.9	52.8	0.117	5.18 / 1.22 / 0.63	0.219	0.639
OLMo-2-7B-SFT, no template	50.9	55.3	0.136	2.26 / 2.88 / 1.70	0.164	0.487
OLMo-2-7B-SFT	51.6	53.3	0.133	2.05 / 2.95 / 1.11	0.182	0.492
OLMo-2-7B-DPO	51.9	53.3	0.128	2.62 / 5.64 / 2.60	0.198	0.488
OLMo-2-7B (RLVR)	51.9	53.2	0.125	2.63 / 5.80 / 2.86	0.212	0.482
SmolLM3-3B base, no template	44.8	46.8	0.130	0.59 / 1.38 / 0.39	–	–
SmolLM3-3B mid-training, no template	42.3	41.5	0.155	1.35 / 6.27 / 11.18	–	–
SmolLM3-3B SFT, no template	45.9	44.3	0.129	1.21 / 2.28 / 10.18	–	–
SmolLM3-3B APO, no template	48.4	51.6	0.155	1.29 / 2.83 / 4.39	–	–
SmolLM3-3B final (merge), no template	50.2	52.7	0.158	1.17 / 2.75 / 3.85	–	–
SmolLM3-3B mid-training	39.9	37.5	0.133	1.37 / 6.23 / 11.56	–	–
SmolLM3-3B SFT	52.0	55.3	0.110	1.00 / 2.74 / 1.27	0.280	0.480
SmolLM3-3B APO	53.9	57.9	0.117	1.30 / 5.15 / 3.03	0.226	0.435
SmolLM3-3B final (merge)	55.2	59.6	0.111	1.20 / 4.66 / 2.56	–	–
Jev 1.13.0 (API)	73.3	83.5	0.113	– / – / –	0.081	0.046

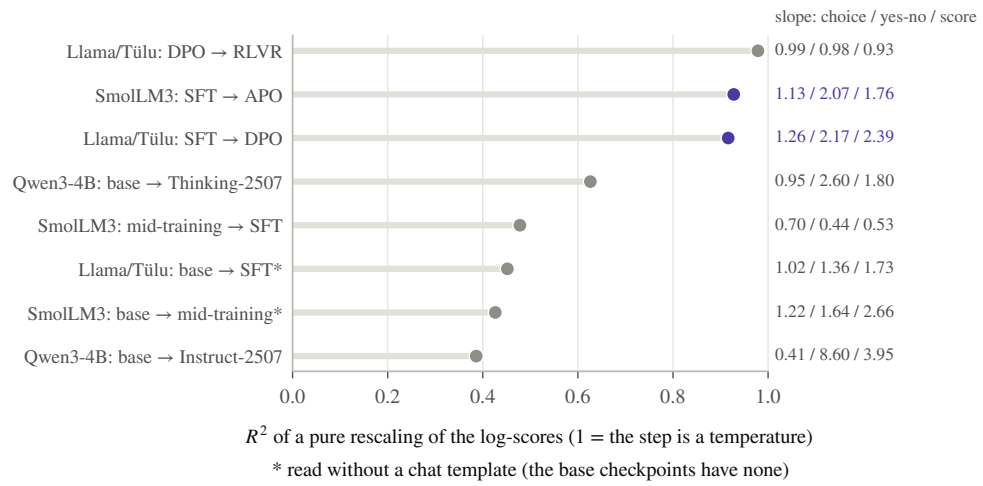


Figure 8: **Is a post-training stage a temperature?** R^2 of a pure rescaling of the earlier checkpoint's log-scores, with the slope per question type (right; above 1 = sharper). Both preference steps (violet) are close to pure rescalings; supervised fine-tuning reorders answers.

D READOUT FINE-TUNING IN FULL

Table 4 gives every fine-tuned model with its starting point. All runs use the same data, prompt and schedule (rank-16 LoRA with $\alpha = 32$ and dropout 0.05 on every attention and MLP projection, AdamW with weight decay 0.01, a one-cycle schedule with peak learning rate 3×10^{-5}, gradient accumulation over 8 families, two epochs, the epoch with the lower validation loss kept). Full fine-tuning keeps fp32 master weights with bf16 autocast and gradient checkpointing; its learning rate was selected at 2B from $\{10^{-5}, 3 \cdot 10^{-6}, 10^{-6}, 3 \cdot 10^{-7}\}$ on validation loss and reused at 4B, where the model is split over two GPUs. Figure 9 shows the reliability of the 9B readout before and after fine-tuning. Confidence intervals in Table 1 come from a paired bootstrap (1,000 resamples) over test items within each task, with the task-averaged metric recomputed on every resample; we report basic (reverse-percentile) intervals, because the calibration error of a single system is biased upward under resampling.

Table 4: Readout fine-tuning from every starting point (one run each; seed spreads in Section 5). Last rows: two other ways to train Qwen3.5-4B for decisions. †Tev1-4B (Together AI, 2026), fine-tuned by its authors to emit the answer letter (37,840 questions), read in its own prompt format with our recipe. ‡An earlier variant of ours: residual decision heads on the readout, trained jointly with a rank-16 LoRA on the same 5,885 questions but with an earlier recipe (for example, without option shuffling), so it is not a controlled ablation of the heads.

Start	Arm	Acc. (%)	Held-out (%)	ECE	TVD	Sure loss	Order flip
Qwen3.5-2B	untuned	57.7	63.4	0.089	0.458	0.212	0.271
	LoRA	70.1	73.6	0.051	0.324	0.339	0.104
	LoRA + coherence	70.2	74.2	0.054	0.315	0.031	0.094
	full	70.3	74.6	0.056	0.315	0.344	0.113
	full + coherence	70.4	74.2	0.056	0.307	0.028	0.109
Qwen3.5-2B-Base	untuned	53.3	57.6	0.124	0.491	0.377	0.402
	LoRA	69.6	72.6	0.053	0.342	0.362	0.101
	LoRA + coherence	70.2	74.2	0.051	0.320	0.034	0.079
Qwen3.5-4B	untuned	66.2	71.9	0.090	0.438	0.151	0.140
	LoRA	74.3	77.4	0.059	0.326	0.283	0.072
	LoRA + coherence	75.1	79.2	0.058	0.303	0.029	0.057
	full	75.2	79.4	0.061	0.322	0.316	0.068
	full + coherence	75.7	80.2	0.056	0.303	0.024	0.064
Qwen3.5-4B-Base	untuned	64.7	71.0	0.095	0.456	0.188	0.191
	LoRA	74.1	76.9	0.064	0.324	0.271	0.064
	LoRA + coherence	74.7	78.3	0.058	0.314	0.033	0.058
Qwen3.5-9B	untuned	68.9	74.5	0.092	0.423	0.141	0.118
	LoRA	75.9	79.8	0.052	0.314	0.323	0.049
	LoRA + coherence	76.3	80.4	0.056	0.301	0.022	0.051
Gemma-4-E4B-it	untuned	65.8	74.2	0.092	0.434	0.205	0.104
	LoRA	73.7	79.2	0.057	0.316	0.295	0.091
	LoRA + coherence	74.2	79.9	0.053	0.297	0.031	0.082
Gemma-4-E4B	untuned	48.8	55.4	0.155	0.441	0.435	0.457
	LoRA	70.8	74.8	0.061	0.335	0.367	0.116
Tülu-3-8B-SFT	untuned	60.3	64.4	0.099	0.448	0.200	0.311
	LoRA	70.8	73.9	0.070	0.321	0.317	0.098
	LoRA + coherence	72.2	75.8	0.064	0.318	0.030	0.090
Tülu-3-8B-DPO	untuned	60.7	63.7	0.110	0.451	0.271	0.308
	LoRA	70.4	73.9	0.072	0.328	0.319	0.104
	LoRA + coherence	70.8	74.4	0.064	0.314	0.033	0.096
OLMo-2-7B-SFT	untuned	51.6	53.3	0.133	0.496	0.182	0.492
	LoRA	70.5	71.7	0.072	0.347	0.271	–
OLMo-2-7B-DPO	untuned	51.9	53.3	0.128	0.509	0.198	0.488
	LoRA	70.3	71.3	0.074	0.350	0.294	–
SmolLM3-3B SFT	untuned	52.0	55.3	0.110	0.461	0.280	0.480
	LoRA	70.5	73.7	0.058	0.350	0.326	–
	LoRA + coherence	71.0	75.6	0.054	0.318	0.039	–
SmolLM3-3B APO	untuned	53.9	57.9	0.117	0.460	0.226	0.435
	LoRA	70.7	73.8	0.058	0.336	0.294	–
	LoRA + coherence	70.7	74.8	0.055	0.319	0.040	–
Jev 1.13.0 (API)		73.3	83.5	0.113	0.432	0.081	0.046
Tev1-4B†	letters, SFT	70.3	76.6	0.086	0.394	0.239	0.100
Qwen3.5-4B	heads + LoRA‡	73.4	76.5	0.096	0.337	0.355	–

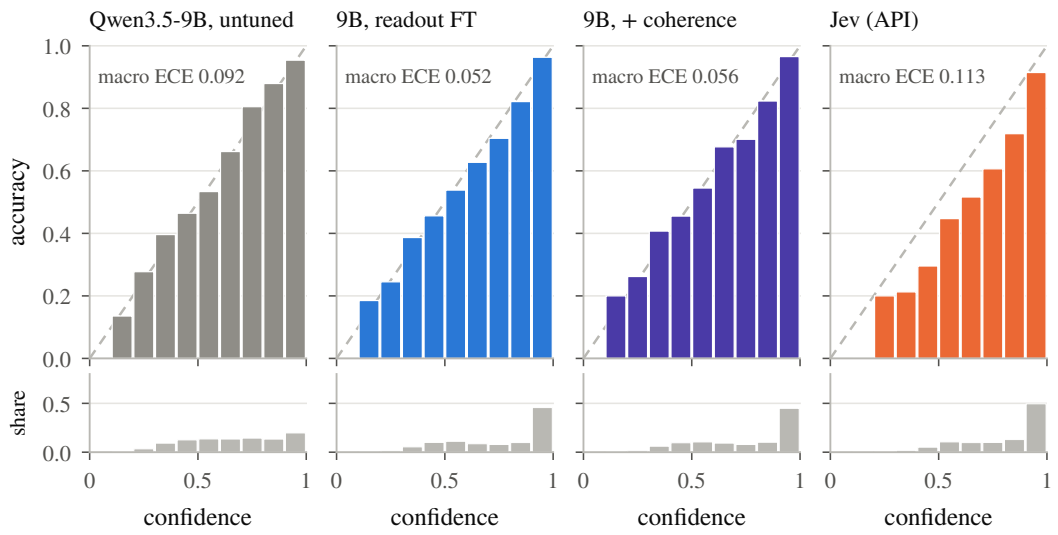


Figure 9: **Sharper and calibrated.** Pooled top-answer reliability over all test items for Qwen3.5-9B before and after readout fine-tuning, and for Jev; the strip shows the share of items per confidence bin. The fine-tuned readouts put about 45% of items in the top bin and stay on the diagonal; Jev is as sharp but over-confident in every bin.

E Ablations of the coherence penalty

Families chosen without the label. For a choice question with more than four options (and ARC-Challenge, whose questions have up to five), a family asks about the correct option and three random others, so which siblings exist depends on the label. Every other family is built without it: every option of a smaller choice, the negation and restatement of a yes/no question, randomly drawn thresholds of a score. Table 5 recomputes the sure loss over the sixteen tasks whose families are label-independent. The penalty's reduction is larger there, not smaller.

Table 5: Sure loss over all 22 tasks, over the 16 whose families are chosen without the label, and over the other 6 (the runs of Table 4).

Model	Arm	All 22	16 label-independent	6 label-dependent
Qwen3.5-9B	untuned	0.141	0.156	0.100
	readout FT	0.323	0.344	0.267
	+ coherence	0.022	0.016	0.038
Qwen3.5-4B	untuned	0.151	0.146	0.165
	readout FT	0.283	0.313	0.204
	+ coherence	0.029	0.022	0.049
Jev 1.13.0 (API)		0.081	0.085	0.071

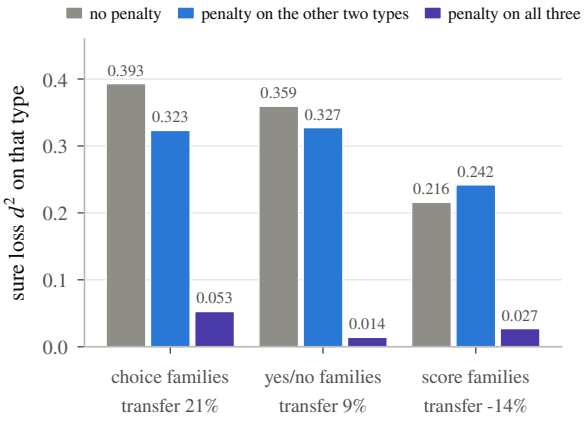


Figure 10: **The penalty is specific to the relations it is trained on** (Qwen3.5-2B). For each family type, the sure loss of its test families with no penalty (three seeds), with the penalty on the other two types only (two seeds), and with the penalty on all three (three seeds). The partial transfer to choice families comes from their yes/no members, which the other types' penalty trains. At 4B the transfer is -28%, 10% and 44% for score, yes/no and choice families (two runs each).

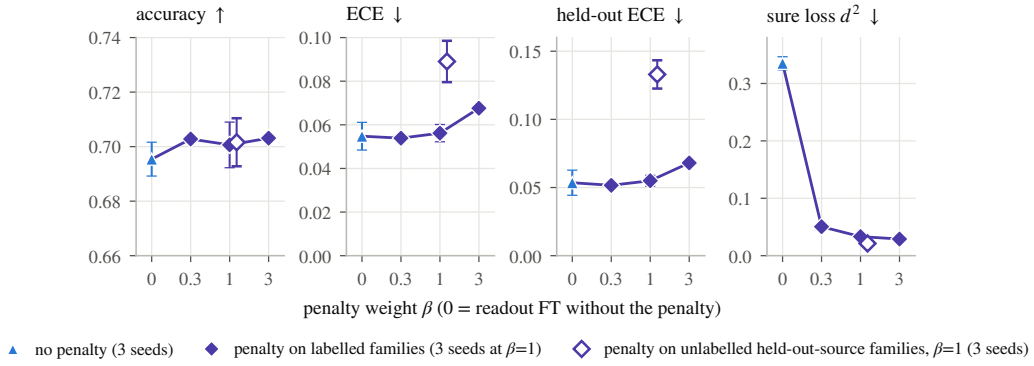


Figure 11: **Penalty weight** (Qwen3.5-2B; $\beta = 0$ and $\beta = 1$ are means over three seeds, ± 1 sd). Accuracy is flat, the sure loss saturates by $\beta \approx 1$, and calibration is unaffected until $\beta = 3$. The hollow marker applies the penalty to unlabelled families from the held-out tasks: they become the most coherent of all, and their calibration error more than doubles.

F Out of distribution

Table 6 reports three behaviours DECIDE-22 does not test: applying a rule stated in the question (a condition, given with the question, that decides the answer; 885 fact patterns from ten rule-application tasks of LegalBench, Guha et al. 2023), choosing "none of the above" when the correct option has been removed (MMLU and ARC-Challenge test items with the option added, the correct one removed in half), and the share of answers that follow an instruction injected into the state (800 test items from SMS Spam, Civil Comments, BoolQ and FEVER, each with and without a planted sentence naming the wrong answer; lower is better). Every readout fine-tune follows injected instructions far less; the effects on the other two are mixed and, at 4B, differ between seeds, so we draw no conclusion from them.

Table 6: Out-of-distribution behaviour (%).

Model	Arm	Stated rule ↑	"None" when gone ↑	Hijacked ↓
Qwen3.5-2B	untuned	60.2	85.0	26.9
	readout FT	53.6	47.2	5.5
	+ coherence	54.5	59.6	4.1
Qwen3.5-4B	untuned	61.9	48.4	39.4
	readout FT	74.2	61.4	11.6
	+ coherence	72.3	68.2	8.9
Qwen3.5-4B, seed 1	readout FT	67.2	67.0	16.4
	+ coherence	62.7	64.2	8.6
Qwen3.5-9B	untuned	68.9	67.4	29.6
	readout FT	80.0	59.2	14.6
	+ coherence	78.2	61.8	5.9
Gemma-4-E4B-it	untuned	74.4	38.2	25.5
	readout FT	79.5	49.8	13.5
	+ coherence	79.1	54.0	12.2
Jev 1.13.0 (API)		92.4	74.4	20.5

A benchmark written by others. The PhishNChips dataset (AreLit, 2026), in the evaluation built on it by anisselbd (2026), asks whether each of 2,000 emails (1,000 of them phishing) is phishing; nothing was fitted on it, and every readout uses the recipe fitted on DECIDE-22 validation. Table 7 gives the verdict question. Every fine-tune calls almost every email legitimate, yet ranks the emails about as well as the untuned model (AUROC): the decision threshold moved, not the ranking. A deployment can repair a threshold with a few labels of its own. We fit one additive shift of the phishing log-odds by maximum likelihood on 16 labelled emails (half phishing) and score every other email, over 200 random draws of the 16. The shift brings every fine-tune to within three points of its untuned model or above it, and cannot help Jev, whose ranking is the weakest.

Table 7: PhishNChips verdict question (%). AUROC ranks the emails by the model's phishing probability; the last columns score the emails not used for the fitted shift, averaged over 200 draws of the 16 labelled emails.

			As served		16 labelled emails	
Model	Arm	AUROC	Accuracy	Recall	Accuracy	Recall
Qwen3.5-4B	untuned	78.4	70.0	43.6	71.6	64.1
	readout FT	86.4	51.8	3.6	75.6	66.4
	+ coherence	92.9	50.6	1.3	83.0	77.0
Qwen3.5-9B	untuned	75.3	66.5	41.1	68.8	62.4
	readout FT	78.8	53.3	6.6	71.6	65.7
	+ coherence	72.3	54.5	9.2	66.6	61.1
Jev 1.13.0 (API)		68.8	62.8	43.6	61.3	60.9

The held-out rating scale. Table 8 asks the same question of Yelp, the held-out task where readout fine-tuning lowers accuracy (Qwen3.5-9B, first run of each arm, 1,000 test reviews). The fine-tuned readouts rank the reviews slightly better than the untuned one and land within one star more often; their loss is in where the star boundaries fall (they predict a mean level of 2.3–2.4 against a true 2.1). One additive bias per level, fitted by maximum likelihood on 16 labelled reviews (stratified by level) and scored on the rest over 200 draws, brings them back to the untuned model's accuracy; the same shift gives the untuned model three points. Jev ranks the reviews better than any readout, and that, not a boundary, is its lead.

Table 8: Yelp star ratings, held out from fine-tuning (%, and Spearman correlation of the expected level with the true one). Last column: one bias per level fitted on 16 labelled reviews, scored on the others (mean of 200 draws).

Model	Accuracy	Within one star	Spearman	16 labelled reviews
Qwen3.5-9B, untuned	53.7	91.6	0.868	56.8
+ readout fine-tuning	47.3	93.1	0.888	53.5
+ coherence penalty	46.1	92.4	0.885	53.2
Jev 1.13.0 (API)	68.5	98.8	0.933	66.9

The held-out tasks, per task. Table 9 breaks down the held-out accuracy of Table 1. Yelp accounts for 3.2 of the 3.7 points by which Jev leads; on the other five the fine-tuned model is behind on four and ahead on hate-speech severity, whose accuracy also varies most between seeds (53.0–65.8%).

Table 9: Accuracy (%) on the six tasks held out from fine-tuning. Fine-tuned columns are means over seeds 0, 1 and 2 of Qwen3.5-9B.

Task	Untuned	Readout FT	+ coherence	Jev	+ coherence − Jev
CLINC150	81.6	85.7	85.8	89.3	−3.5
ARC-Challenge	93.6	94.7	94.7	97.9	−3.2
Yelp reviews	53.7	46.8	49.5	68.5	−19.0
Measuring Hate Speech	41.6	59.8	59.1	52.7	+6.4
FEVER (gold evidence)	94.3	96.6	96.5	97.2	−0.7
StrategyQA (with facts)	82.0	93.6	93.5	95.6	−2.1
Mean	74.5	79.6	79.9	83.5	−3.7

Calibration controls. The open readouts' recipe is fitted on validation labels, Jev is served as is, and the recipe of Table 1 sees the held-out tasks' validation items. Table 10 removes both advantages: Jev gets the same recipe, fitted on the same validation items (4,120 new API calls), and each recipe is also refitted on the sixteen fine-tuning tasks alone, so that the held-out tasks are scored by a model and a recipe that never saw a label of theirs. Brier score is given beside ECE. The fine-tuned model is better calibrated overall and on the fine-tuning tasks in every setting; on the held-out tasks Jev with the same recipe is better.

Table 10: Calibration error and Brier score (ECE / Brier) under each recipe. The recipe is refitted on the validation items of all tasks, or of the sixteen fine-tuning tasks only; macro averages over tasks.

System	Recipe fitted on	16 fine-tuning tasks	6 held-out	All 22
9B + coherence, seeds 0–2	all tasks	0.048 / 0.321	0.070 / 0.248	0.054 / 0.301
9B + coherence, seed 0	all tasks	0.050 / 0.318	0.073 / 0.249	0.056 / 0.299
	16 fine-tuning tasks	0.043 / 0.317	0.083 / 0.255	0.054 / 0.300
Jev 1.13.0 (API)	none (as served)	0.122 / 0.391	0.090 / 0.235	0.113 / 0.349
	all tasks	0.071 / 0.366	0.061 / 0.221	0.068 / 0.326
	16 fine-tuning tasks	0.075 / 0.366	0.074 / 0.228	0.075 / 0.328

G Decoding protocol

The decode baselines of Section 5.1 use the readout's own rendering of each question, with the answer cue replaced by an instruction to end with two lines, "Answer: *one of the allowed answers*" and "Confidence: *the probability, from 0 to 100, that your answer is correct*". Without thinking the model decodes greedily (at most 256 tokens); with thinking it samples with the model card's recommended settings (temperature 0.6, top-p 0.95, top-k 20) under an 8,192-token budget. The answer is parsed from the last "Answer:" line after the reasoning; unparseable answers count as

wrong with confidence $1/K$. To read after thinking, the reasoning is closed if the budget ran out, the answer cue is appended, and the allowed answers' probabilities are read at the cue as in Equation 1. The sample is 100 test items per task (2,200 in total, seed 0), scored against every readout on exactly the same items. Latency is the median over 30 of these items at batch size one with vLLM on one A40, for the prefill plus one token (the readout), for the direct decode and for the thinking decode.

The same at 4B. On the same items, Qwen3.5-4B decoding matches the readout's accuracy (66.2% against 66.6%) with a far less informative confidence (AUROC 0.62 against 0.74), and reasoning adds a single point while a fifth of its reasoning runs exceed the 8,192-token budget; the fine-tuned 4B readout reaches 75.0%.

H Further related work

Readouts as classifiers. Reading a class from the probabilities of label words is how pattern-based few-shot learners use a language model (Schick & Schütze, 2021; Gao et al., 2021); they fine-tune that readout with cross-entropy for few-shot accuracy, where we fine-tune it with proper scoring rules for calibration and coherence on typed decisions with up to 151 options. An earlier variant of ours that added trained decision heads to the readout did not do better (Table 4), but it differs from readout fine-tuning in more than the heads, so we draw no conclusion about heads as such.

Calibration after fine-tuning. Fine-tuned pretrained transformers are fairly well calibrated in domain and less so out of domain (Desai & Durrett, 2020); fine-tuning and post-hoc adjustment improve the calibration of question-answering models (Jiang et al., 2021), and fine-tuning on a small set of graded answers yields uncertainty estimates that generalize across tasks (Kapoor et al., 2024). Verbalized confidence is over-confident across models and elicitation strategies (Xiong et al., 2024); Section 5.1 finds the same for decisions, before and after reasoning.

Consistency as a test. Consistency checks can evaluate models whose answers cannot be verified directly (Fluri et al., 2024). The sure loss is a quantitative check of this kind with a guarantee attached: any violation means a set of answers exists that scores better in every world (Equation 3), and the check needs no labels, so it applies to unlabelled inputs at deployment.