

# Releasing the Inner Jev: A Simple Recipe for Open Decision Models

Jonathan Geuter<sup>\*12</sup> Andrew Cheng<sup>\*1</sup> David Alvarez-Melis<sup>12</sup>

## Abstract

LLM-based decision-making systems, also called System One models, have recently gained popularity, most notably through Jev (Almeida, 2026). Given a typed input state, a question, and a set of options, they return a probability for each option in a single forward pass, without text generation. We show that existing open instruction-tuned language models can achieve comparable performance *without any additional training*, by reading out answers from letters (A, B, C, etc.) in the model’s output logits at the last token position. Furthermore, we show how a simple supervised fine-tuning recipe can improve their performance further. We evaluate four open instruction-tuned language models: Gemma 4 31B, Gemma 4 12B, Qwen3.6-27B, and Qwen3.5-4B. We fine-tune each using LoRA and a decision head initialized from a slice of the language model head. Furthermore, we introduce *OpenDecisions*, a novel 2,352-question validation dataset designed to reflect the task composition of JevBench (JevBench maintainers, 2026) and which computes a Capability score based on accuracy and calibration metrics. Without fine-tuning, Gemma 4 31B scores 79.6 on OpenDecisions, compared to Jev’s 80.1. After undergoing our finetuning recipe, it scores 84.4 on OpenDecisions and 81.5 on JevBench v1.7, outperforming Jev by 4.3 and 4.4 Capability points respectively and setting a new state-of-the-art on OpenDecisions. On a suite of public decision datasets, our models achieve accuracy comparable to Jev, with lower calibration error. Our results show that open language models already possess System One capabilities, and with simple fine-tuning they outperform Jev on a wide range of decision tasks. We release our models, serving code, and OpenDecisions.

 Jeff 1.0 models  jeff-serve  OpenDecisions

<sup>\*</sup>Equal contribution <sup>1</sup>Harvard University <sup>2</sup>Kempner Institute. Correspondence to: Jonathan Geuter <jonathan.geuter@gmx.de>.

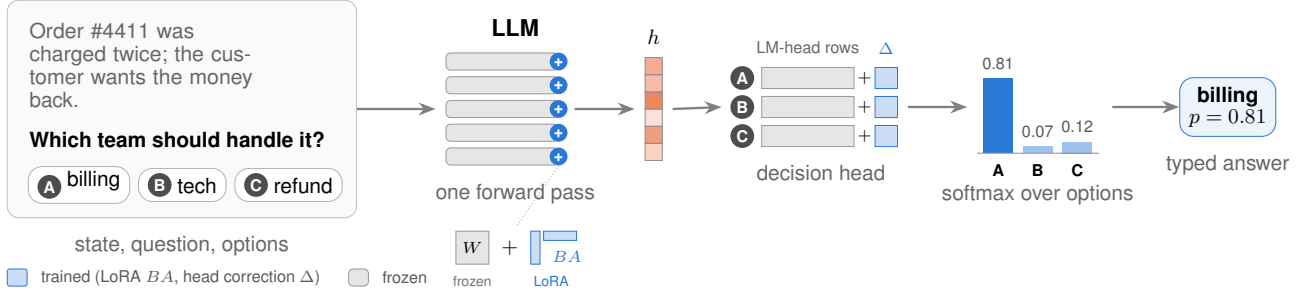
## 1. Introduction

Many applications require choosing among a fixed set of options, such as routing a support ticket, checking a request against a policy, or picking the right tool in an agentic workflow. Language models can make these decisions by generating text, but generating tokens sequentially introduces additional inference costs and latency. For tasks requiring only a choice and its associated probabilities, a single forward pass avoids this overhead and provides uncertainty estimates that downstream systems can use directly.

Using a pretrained language model with a classification head has long been well-established in the literature, including ULMFiT (Howard & Ruder, 2018), GPT (Radford et al., 2018), and BERT (Devlin et al., 2019). Jev (Almeida, 2026) adopts this approach: given a state, a question, and a set of options, it returns a probability for each option. TypeSafe, the company behind Jev, describes it as using a specialized architecture, a parallel sampler, and a training method called Reinforcement Learning for Calibrated Decisions (RLCD) (Almeida, 2026). Its weights and training recipe are not public. The format has attracted several open implementations, which are compared on the JevBench leaderboard (JevBench maintainers, 2026).

*Are specialized architectures or training methods necessary for competitive decision models?* We find that a simple approach using open-source language models already achieves performance comparable to Jev. When given a multiple-choice question, we extract probabilities over the options from the model’s logits (see Section 4 for details), an established approach to evaluating multiple-choice models (Hendrycks et al., 2021a; Robinson & Wingate, 2023), and apply simple temperature scaling for calibration. Without further fine-tuning, Gemma 4 31B (Gemma Team, Google DeepMind, 2026) and Qwen3.6-27B (Qwen Team, 2026b) achieve Capability scores of 79.6 and 79.0 on our validation set *OpenDecisions* respectively, compared with 80.1 for Jev. Both models also achieve similar average accuracy to Jev across seven publicly available decision datasets.

Next, we investigate whether supervised fine-tuning can improve performance further. We add LoRA adapters to the language model and initialize a decision head from the open-source model’s slice of the language model head cor-



**Figure 1. A Jev-style decision in one forward pass.** The request is rendered as a multiple-choice prompt with lettered options. The LLM reads it once; the decision head maps the last hidden state  $h$  to one logit per option, starting from the language-model head’s rows for the letters, so the untrained model is the zero-shot readout. Training adds LoRA adapters and a learned correction  $\Delta$  to the head. A softmax over the real options gives the probabilities.

responding to the answer letter tokens (A, B, C, etc.). Both are trained jointly using cross-entropy on questions drawn from a pool of 417,825 public and language-model-labeled examples. After 400 training steps (about five GPU-hours per run), Gemma 4 31B improves from 79.6 to 84.4 Capability, averaged over three runs. This exceeds Jev by  $4.3 \pm 2.5$  points and sets a new state-of-the-art on OpenDecisions. These results show that existing language models can match and exceed Jev on decision tasks using a short supervised training procedure, without a specialized architecture or reinforcement learning.

Our contributions are:

- **Existing language models are competitive decision models.** We show that without finetuning, open instruction-tuned language models achieve comparable performance to Jev. This holds on both OpenDecisions and seven publicly available decision datasets (Section 5.1).
- **A validation set for decision models.** We introduce OpenDecisions, a validation dataset for decision models containing 2,352 questions across 11 task families, three question types, four difficulty levels and 23 languages. Its labels are checked programmatically or by a panel of three models, and its scores broadly reproduce the rankings of systems on the official JevBench leaderboard (Section 3).
- **A simple training recipe for decision models.** Our short supervised LoRA training procedure improves Gemma 4 31B by  $4.8 \pm 2.0$  Capability points on OpenDecisions, exceeding Jev and setting a new state-of-the-art. Ablations show that adapting the language model drives most of this improvement, while training the decision head alone provides little to no benefit (Sections 4, 5.2 and 5.4).

## 2. Background and related work

**Jev-style decision models.** A Jev-style decision consists of a state (text or JSON), a question, and  $K$  possible options. JevBench considers three types of questions: *choice* (pick one of  $K$  options), *yes/no* (do the given criteria hold for the state?), and *score* (pick one of several ordered levels). For yes/no questions, the prediction is represented by a single probability  $P(\text{yes})$ . These probabilities allow downstream systems not only to select an answer, but also to return a confidence associated with it. For this to be reliable, the predicted probabilities should be well-calibrated.

**Classification with pretrained LLMs.** LLM classification predates Jev: ULMFiT adds a classification head to a pretrained recurrent language model, while GPT and BERT adapt pretrained transformers using supervised output layers (Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2019). PET (Schick & Schütze, 2021) uses cloze-style prompts to construct soft labels and then trains a conventional sequence classifier with a classification head. A separate line of work handles labels supplied in natural language: Yin et al. (2019) formulate zero-shot classification as entailment between the input and candidate label hypotheses, while GLiClass (Stepanov et al., 2025) jointly encodes the input and candidate labels and scores them in a single pass.

**Open decision models.** Several open systems implement this format using existing language models. Some require no additional training and obtain probabilities directly from answer-letter logits, including SemIf (SemIf contributors, 2026), open-alternative-jev (IkerMoel, 2026), Cygnet (blockbrain-ai, 2026) and deck31b (krishnagogineni-765, 2026); others average over option orders (Zhang et al., 2026) or score each option in its own prompt (Yinsongxu, 2026). Fine-tuned systems include Plumb (crh225, 2026), JevK5 (allebee, 2026), Hopper (HopitAI, 2026) and reflex (kshetrajna12, 2026) (LoRA on Qwen3.5-4B), decider (Marosi, 2026) (full fine-

tuning on about 95 datasets), Winnow-12B (EldanRing, 2026), Torchcast-decision-12B (torchcast-ai, 2026), Quyet-1.0 (Nguyen, 2026) (LoRA on Gemma 4 31B), and several fine-tunes of Qwen3.8-27B: Eikos (Vicentino, 2026), Auto-Jev (denis-pplx, 2026), Clef (Cloudflare, 2026) and Open-Jev (Cai, 2026). Despite the variety of approaches, few of these systems report evaluations beyond the JevBench leaderboard.

**Concurrent work.** Li & Wagle (2026) also study whether existing language models can perform Jev-style decisions without additional training. Rather than using answer letters, they extract probabilities from numbered options using a partially completed answer prompt. They find that an unmodified Qwen3.5-4B performs comparably to other decision models built on the same backbone, which corroborates our zero-shot finding. They also study fine-tuning 4B and 0.6B models with a listwise loss with KL regularization. We compare their number readout with our letter readout on OpenDecisions and find similar zero-shot Capability scores for Gemma 4 31B (80.5 for numbered options and 79.6 for answer letters). Since the number readout requires multiple forward passes (each digit requires a forward pass, plus one to denote the end of the number), whereas our letter readout requires only one, we adopt the latter.

**Calibration.** Correct predictions alone are not enough when downstream systems rely on the reported probabilities. Neural networks are often overconfident, and temperature scaling provides a simple way to improve calibration without changing the preferred answer (Guo et al., 2017; Desai & Durrett, 2020). Previous work shows that pretrained language models can produce reasonably calibrated probabilities, while instruction tuning can increase overconfidence (Kadavath et al., 2022; Tian et al., 2023). Predictions can also depend on the order and labels of the answer options (Zheng et al., 2024; Pezeshkpour & Hruschka, 2023). In this work, we show that the model output probabilities themselves already lead to better calibration than Jev (see Section 5). We leave more sophisticated calibration approaches for future work.

### 3. OpenDecisions: a validation set for Jev-style models

**Why a new dataset.** Evaluating decision models requires (i) enough questions to reliably compare their performance and (ii) sufficient coverage of the tasks they are designed to solve. Existing public datasets have limitations in both respects. JevBench has released only 231 questions, which we call JevBench-Open. Other public decision datasets, such as those from SemIf and TypeSafe, contain only 100 to 350 questions as well. Although larger multiple-choice benchmarks exist, they primarily test general knowledge

*Table 1. OpenDecisions families and label sources.* Programmatic items are written and labeled by code; authored items are written by an LLM; real requests are user requests from public datasets. Calibration: questions with exact answer probabilities.

| Family                 | Code | Authored | Real | Total |
|------------------------|------|----------|------|-------|
| LLM routing            | 22   | 391      | 281  | 694   |
| Multilingual           |      | 393      | 29   | 422   |
| Fairness pairs         |      | 236      |      | 236   |
| Business use cases     | 20   | 193      |      | 213   |
| Counting traps         | 162  |          |      | 162   |
| Negation pairs         | 160  |          |      | 160   |
| Option order           |      | 120      |      | 120   |
| Perturbation pairs     |      | 110      |      | 110   |
| Composite rules        | 106  |          |      | 106   |
| Calibration, generated | 94   |          |      | 94    |
| Calibration            | 35   |          |      | 35    |
| Total                  | 599  | 1,443    | 310  | 2,352 |

rather than decisions based on supplied information. To address these limitations, we introduce OpenDecisions, a 2,352-question validation set designed to reflect the task composition of JevBench.

**Coverage.** OpenDecisions has 2,352 decisions in 11 families (Table 1): routing requests to models or teams, multilingual decisions, fairness pairs, business use cases, counting and arithmetic traps, negation pairs, option-order and perturbation pairs, composite policy rules, and questions with exact answer probabilities. It has all three question types (1,178 choice, 586 yes/no and 588 score questions), four difficulty tiers (easy, standard, judge, hard) and 23 languages (78% English). Its mix of types, tiers, families and languages follows the published composition of the public JevBench v1.6 split.

**Labels.** The items come from three sources. 599 are written and labeled by code. 1,443 were written by LLM authoring agents (Claude Opus 4.8 and 5.5; Anthropic, 2026) from fixed briefs and then confirmed by a panel of three open models (Qwen3.8-27B, gpt-oss-120b and GLM-4.7-Flash; Qwen Team, 2026c; OpenAI, 2025; Z.ai, 2026) that answer each item blind, with reasoning. An item was kept only if all three agreed with the author’s answer, or two of three did and their average probability on that answer was at least 0.5. 310 are real user requests (Chatbot Arena (Chiang et al., 2024), Dolly (Conover et al., 2023), GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021b; Lightman et al., 2023)), again answered by the three-model panel, and kept only when all models agree.

**Quality checks.** Some questions are constructed in pairs, such as the same question with its answer options reordered or with a detail of the input changed. We remove both questions if either fails verification by the three-model panel.

We also compare candidate questions against our training data and the publicly released JevBench questions to check for overlap, removing 43 samples.

**Validity.** 14 public decision models come out on OpenDecisions in nearly the order of JevBench (JevBench maintainers, 2026) (Spearman correlation of 0.90). Four of these systems were never used while building OpenDecisions; a linear fit on the other ten predicts their leaderboard scores with an error of 3.0 points. JevBench-Open and a validation set drawn from our training distribution predict the leaderboard much worse (Appendix C).

**Scoring.** We score a system by  $\text{Capability} = (I + C)/2$ , adapted from JevBench’s scorer without its abstention band for yes/no answers. Intelligence  $I$  is chance-corrected accuracy,  $100(\text{acc} - \text{chance})/(1 - \text{chance})$ , computed per question type and difficulty tier and averaged with tier weights 0.1, 0.2, 0.3 and 0.4 (easy to hard) and equal type weights. A yes/no answer counts as correct when its more likely side is correct. Calibration  $C$  averages one score per type:  $100(1 - \text{ECE}/0.5)$  (Pakdaman Naeini et al., 2015) of the top answer on hard choice items, of  $P(\text{yes})$  on yes/no items, and for score items the mean of the top-answer ECE score and  $100(1 - \text{RPS})$ , the ranked probability score (Epstein, 1969). We report 95% intervals as  $\text{value} \pm 1.96 \text{ SE}$ . The standard error SE combines question sampling (a paired bootstrap over question groups, 1,000 resamples, the same for all systems) and, for trained models, the variation between training seeds.

## 4. A simple recipe

In this section, we explain how decision models work in detail, and give a simple training recipe to apply on pretrained LLMs to improve their decision-making capabilities.

**Letter readout.** Each sample is presented to the model with options labeled A, B, C, and so on. Following SemIf (SemIf contributors, 2026), we use a JSON message containing the state (evidence), question (criterion), and answer options. The model is instructed to select one answer letter, using its standard chat template with thinking disabled (see Appendix A for details).

Rather than generating a textual answer, we extract the model’s logits for the answer letters after a single forward pass. Let  $h \in \mathbb{R}^d$  denote the final hidden state at the last prompt position, after the final normalization layer, and let  $W_0 \in \mathbb{R}^{16 \times d}$  contain the rows of the language-model head corresponding to the letters A through P. For a question with  $K$  options, we obtain the probability of option  $k$  by

applying a softmax over these  $K$  logits:

$$p_k = \frac{\exp((W_0 h)_k)}{\sum_{j=1}^K \exp((W_0 h)_j)}, \quad k = 1, \dots, K. \quad (1)$$

This converts the scores for the valid answer letters into a probability distribution over the available options. Each letter is represented by a single token, so all probabilities can be obtained in one forward pass. For Gemma, we retain its original final-logit soft-capping before applying softmax, as described below. Yes/no questions are represented by two options, while scoring questions use ordered answer levels. The readout supports up to 16 options.

**Decision head.** Rather than learning a new decision head from scratch, we exploit the pretrained language model’s answer-letter readout and add a trainable correction:

$$z = W_0 h + A \text{std}(h) + b, \quad (2)$$

followed by the softmax over the  $K$  options. Here,  $W_0$  are the fixed letter rows from Eq. (1),  $\text{std}(h)$  standardizes  $h$  with mean and standard deviation computed once on training data before training (which are kept fixed during training), and the learned correction  $A \in \mathbb{R}^{16 \times d}$ ,  $b \in \mathbb{R}^{16}$  starts at zero. For Gemma, we preserve the model’s original final-logit soft-capping by replacing  $W_0 h$  with  $c \tanh(W_0 h/c)$  with  $c = 30$ . Without soft-capping, the decision head is a single affine function of  $h$ . After training, the LoRA adapter can be merged into the pretrained weights, so inference requires only one forward pass.

**Calibration.** The probabilities produced by language models are not always well calibrated (Guo et al., 2017; Desai & Durrett, 2020; Tian et al., 2023). In our experiments, Gemma’s probabilities are particularly overconfident. We address this using temperature scaling (Guo et al., 2017), which adjusts the predicted probabilities without changing the preferred answer. We fit one temperature for each question type by minimizing negative log-likelihood on 3,884 held-out questions drawn from the training distribution. These questions use the same format as OpenDecisions, but no OpenDecisions questions are used for calibration. We note that this is a very simple calibration strategy, and more sophisticated calibration approaches, such as directly incorporating calibration in the training loss, could likely help.

One challenge is that the calibration data differ from OpenDecisions. As a result, the fitted temperatures do not always transfer well. For the trained Qwen models, temperatures below one sharpen the predictions and reduce performance on OpenDecisions. We therefore restrict temperatures to be at least one, allowing calibration to reduce confidence but not increase it (Appendix G).

**Table 2. Training pool** (417,825 questions). Soft = the target is a distribution (annotator votes or teacher probabilities). Teacher labels come from Qwen3.5-122B-A10B and Qwen3.6-27B.

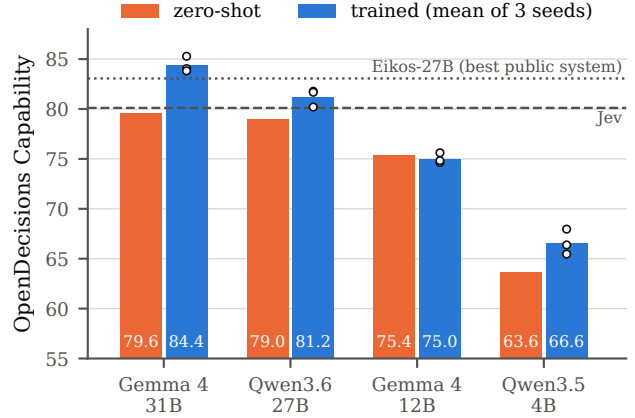
| Part                              | Rows           | Labels              |
|-----------------------------------|----------------|---------------------|
| decider public mixture (79 tasks) | 300,042        | dataset (hard)      |
| Civil Comments                    | 30,000         | annotators (soft)   |
| Measuring Hate Speech             | 30,000         | annotators (soft)   |
| STS-B                             | 4,760          | similarity (soft)   |
| ChaosNLI                          | 2,886          | 100 votes (soft)    |
| synthetic documents (ours)        | 30,641         | teachers (soft)     |
| decider custom questions          | 19,496         | teachers (soft)     |
| <b>Total</b>                      | <b>417,825</b> | <b>117,783 soft</b> |

**Training data.** We train on a pool of 417,825 questions drawn from public datasets and additional questions labeled by language models (Table 2). The largest portion consists of 300,042 from the public training mixture of decider, covering 79 tasks (Marosi, 2026). We also include four public datasets with soft targets derived from human annotations: Civil Comments (Borkan et al., 2019), Measuring Hate Speech (Kennedy et al., 2020), STS-B (Cer et al., 2017), and ChaosNLI (Nie et al., 2020).

The remaining 50,137 questions have soft labels generated by two LLM teachers, Qwen3.5-122B-A10B and Qwen3.6-27B (averaging two reasoning samples from each model). These include 30,641 questions based on synthetic business documents spanning approximately 40 domains, and 19,496 questions from the decider repository that we relabeled. No OpenDecisions questions are used for training. Appendix C.1 examines exact overlap and input-distribution differences between the training pool and OpenDecisions.

**Training.** We fine-tune the models using supervised learning, keeping the original language-model weights frozen. Specifically, we add rank-32 LoRA adapters ( $\alpha = 64$ , no dropout) to all attention and MLP projections, including the linear-attention projections in the Qwen models, and train them jointly with the decision head. We minimize cross-entropy against the target probability distribution when available, or the correct answer otherwise. We also shuffle the order of the answer options during training.

We use AdamW (Loshchilov & Hutter, 2019) with learning rates of  $2 \cdot 10^{-5}$  for LoRA and  $10^{-4}$  for the decision head. The learning rate increases during the first 250 steps, remains constant until step 300, and then decays to zero at step 400. Each step processes 65,536 padded tokens, corresponding to approximately 190 questions. Over 400 steps, a run processes about 76,000 questions, equivalent to 18% of the training pool. Training Gemma 4 31B takes 78 minutes on four H200 GPUs (about five GPU-hours), compared with 64 minutes for Qwen3.6-27B, 40 minutes for Gemma 4 12B, and 23 minutes for Qwen3.5-4B. We use



**Figure 2. Zero-shot reaches Jev; a short training run goes beyond.** OpenDecisions Capability of the four open models without training (orange) and after training (blue, mean of three seeds; dots: seeds; Section 4). Dashed: Jev. Dotted: the strongest public system (Eikos-27B). The two models of about 30B parameters reach Jev’s level zero-shot, and training moves Gemma 4 31B clearly beyond it.

the same training procedure for Gemma 4 31B, Qwen3.6-27B and Gemma 4 12B. For Qwen3.5-4B, we added three changes that empirically helped this model: a LoRA learning rate of  $10^{-4}$ ; as target, the average of the data target and the soft prediction of our trained Gemma 4 31B; and half of the training questions shown in the format of the evaluation questions. We use no reinforcement learning, reward models, or auxiliary losses. Further details are provided in Appendix B.

## 5. Results

**Setup.** We evaluate four open models: Gemma 4 31B and 12B (Gemma Team, Google DeepMind, 2026), Qwen3.6-27B (Qwen Team, 2026b), and Qwen3.5-4B (Qwen Team, 2026a), both before and after fine-tuning (three training seeds per model). We compare them with Jev, accessed through its API, and 13 public systems using their authors’ implementations. All systems are evaluated on the same questions in OpenDecisions. For each public system, we report the better of its original predictions and predictions recalibrated using our separate calibration set, based on its OpenDecisions score. This favors the competing systems. Since we selected our training recipe using OpenDecisions, we also evaluate on seven held-out decision datasets, JevBench-Open (a small 231-sample public dataset released by JevBench), and six standard multiple-choice benchmarks.

### 5.1. LLMs as zero-shot decision models

We find that strong open language models already achieve performance comparable to Jev without fine-tuning. On

OpenDecisions, Gemma 4 31B achieves a Capability score of 79.6 ( $I = 71.9$ ,  $C = 87.4$ ) and Qwen3.6-27B scores 79.0, compared with 80.1 for Jev (Figure 2, Table 3). Both models also achieve similar average accuracy to Jev across seven publicly available decision datasets (Section 5.3).

Calibration plays an important role in this result. Gemma’s raw answer-letter probabilities are overconfident, and temperature scaling raises its Capability score from 72.9 to 79.6 without changing its preferred answers. The fitted temperatures are 6.0 for choice questions and 7.2 for yes/no questions (Appendix G). The answer-letter readout also discards very little probability mass: Gemma 4 31B assigns 99.99% of its next-token probability to the available answer letters, and every backbone we evaluate assigns over 99.4% on average (Appendix H).

Performance also depends on the underlying model. The smaller Gemma 4 12B and Qwen3.5-4B models score 75.4 and 63.6 on OpenDecisions, respectively. On the public decision datasets, zero-shot accuracy increases rapidly with model size up to 4B parameters, and approaches Jev’s level between 12B and 31B (Appendix E). The Gemma 4 31B base model evaluated using a plain completion prompt scores only 64.0, while the instruction-tuned model scores 78.9 with the same prompt (Appendix H). Instruction tuning therefore matters, and our training does not replace it: the base model reaches only 70.3 after training with our recipe (Section 5.4).

## 5.2. A short training run goes beyond Jev

Our LoRA training raises Gemma 4 31B from 79.6 to 84.4 Capability on OpenDecisions, a gain of  $4.8 \pm 2.0$ . The result is the highest OpenDecisions score of all systems we evaluated (Table 3), leading Jev by  $4.3 \pm 2.5$ , Eikos-27B by  $1.3 \pm 2.0$ , deck31b by  $3.1 \pm 1.9$ , Quyet-1.0-Large by  $2.5 \pm 2.1$ , and AutoJev-27B by  $3.9 \pm 2.4$ . Our trained model has the best Calibration of all systems (92.3); Quyet-1.0-Large has a slightly higher Intelligence (76.7 vs. 76.4) but lower Calibration (86.9).

The smaller models gain less from training than the 31B: With the same recipe, Qwen3.6-27B gains  $2.2 \pm 2.2$  and Gemma 4 12B  $-0.4 \pm 1.8$ . On Qwen3.5-4B the recipe does not help (63.0 vs. 63.6 zero-shot); with the three changes for this model (Section 4) it gains  $3.0 \pm 2.6$ . For Gemma 4 12B, longer training does not gain beyond noise either: up to 3,000 steps it scores between 75.0 and 76.3, against 75.4 zero-shot (Table 9). Training does, however, make its temperatures unnecessary: the raw readout (no temperature scaling) rises from 65.4 to 74.0 (Appendix G). Training raises Intelligence for every model (27B +5.0, 12B +1.4, 4B +6.8), but it lowers Calibration for these three (27B -0.6, 12B -2.2, 4B -0.8), which cancels part of the gain. Two reasons may contribute: we chose the

recipe on the 31B model, and part of the teacher labels come from Qwen3.6-27B itself. On the standard multiple-choice benchmarks, training does raise the accuracy of every model (Section 5.3).

We release our trained 31B as Jeff-1.0-Large, and the other models as Jeff-1.0-Qwen-27B, Jeff-1.0-Medium (Gemma 4 12B), and Jeff-1.0-Qwen-4B.

## 5.3. Public evaluation sets

OpenDecisions was built by us and we used it to select our training recipe. We also evaluate our models on public datasets to assess whether the improvements generalize beyond OpenDecisions.

**Held-out decision datasets.** Table 4 reports accuracy and calibration on seven public decision datasets containing 9,212 questions that were not included in our training data. We evaluate both Jev and our models on the same questions for comparison. Averaged over the seven sets, the trained 31B model scores 89.0, the trained 27B model 89.4 and Jev 88.0; every open model of about 30B parameters, trained or not, is at or above Jev. The largest set, 7,800 items from the held-out tasks of decider, gives a clear margin over Jev:  $2.2 \pm 0.7$  for the trained 31B model and  $1.6 \pm 0.7$  for the trained 27B model. The trained 31B model is also the best-calibrated system on these sets: its mean ECE over the six hard-label sets is 0.047, against 0.066 for Jev. Jev is best on SemIf’s authored set (97.2), which is public and may have been used in its training.

**JevBench-Open.** On the 231 items JevBench released publicly (Table 3), the trained 31B model answers 92.1% correctly and scores 80.1 Capability, against 86.6% and 73.6 for Jev. Without training, Gemma 4 31B already answers 91.8%. Among public systems only deck31b is slightly higher (93.1%, 80.5). The concurrent work of Li & Wagle (2026) reports 81.4% for its training-free and 84.0% for its LoRA-trained 4B model. With 231 items, one system’s Capability has a bootstrap standard deviation of about 3 points, so this set separates strong systems only coarsely.

**JevBench leaderboard.** The official JevBench leaderboard (JevBench maintainers, 2026) (v1.7) measured Jeff-1.0-Large on a set of 1,500 items. It scores 81.5 Capability ( $I = 71.9$ ,  $C = 91.1$ ), second of all listed systems and 0.2 points behind Quyet-1.0-Large (81.7); Jev scores 77.1 (Appendix D, Table 14). Each system is measured on its own draw, so a difference of 0.2 points is within noise. The leaderboard counts a yes/no answer with  $0.2 < P(\text{yes}) < 0.8$  as an abstention, which is scored as wrong; for the submission we therefore moved such answers to just outside the nearer edge (0.801 or 0.199).

**Table 3. Main results.** OpenDecisions Intelligence, Calibration and Capability, and Capability and top-1 accuracy on JevBench-Open (the 231 items JevBench released publicly). Ours: letter readout with fitted temperatures; trained = mean of three seeds of the recipe (Section 4; Qwen3.5-4B with three changes); Jeff-1.0-Large is the trained Gemma 4 31B (we release seed 1). Public systems: as served or with fitted temperatures, whichever scores higher on OpenDecisions. Trained: ✓ fine-tuned, – frozen readout, ? unknown. Bold: best per column.

| System                 | Backbone    | Trained | OpenDecisions |             |             | JevBench-Open |              |
|------------------------|-------------|---------|---------------|-------------|-------------|---------------|--------------|
|                        |             |         | <i>I</i>      | <i>C</i>    | Capability  | Capability    | Accuracy (%) |
| <i>Ours</i>            |             |         |               |             |             |               |              |
| Zero-shot              | Gemma 4 31B | –       | 71.9          | 87.4        | 79.6        | 73.4          | 91.8         |
| Jeff-1.0-Large         | Gemma 4 31B | ✓       | 76.4          | <b>92.3</b> | <b>84.4</b> | 80.1          | 92.1         |
| Zero-shot              | Qwen3.6-27B | –       | 67.5          | 90.5        | 79.0        | 77.1          | 88.3         |
| Trained                | Qwen3.6-27B | ✓       | 72.6          | 89.9        | 81.2        | 78.7          | 88.5         |
| Zero-shot              | Gemma 4 12B | –       | 64.0          | 86.8        | 75.4        | 72.0          | 86.1         |
| Trained                | Gemma 4 12B | ✓       | 65.5          | 84.6        | 75.0        | 77.2          | 87.6         |
| Zero-shot              | Qwen3.5-4B  | –       | 46.1          | 81.2        | 63.6        | 63.4          | 80.1         |
| Trained                | Qwen3.5-4B  | ✓       | 52.8          | 80.4        | 66.6        | 64.7          | 79.7         |
| <i>Public systems</i>  |             |         |               |             |             |               |              |
| Eikos-27B              | Qwen3.8-27B | ✓       | 75.7          | 90.5        | 83.1        | 74.7          | 91.3         |
| Quyet-1.0-Large        | Gemma 4 31B | ✓       | <b>76.7</b>   | 86.9        | 81.8        | 78.8          | 90.9         |
| deck31b                | Gemma 4 31B | –       | 73.6          | 89.0        | 81.3        | <b>80.5</b>   | <b>93.1</b>  |
| AutoJev-27B            | Qwen3.8-27B | ✓       | 70.8          | 90.2        | 80.5        | 77.1          | 87.4         |
| Jev 1.13.0 (API)       | closed      | ✓       | 71.9          | 88.3        | 80.1        | 73.6          | 86.6         |
| Torchcast-decision-12B | Gemma 4 12B | ?       | 68.0          | 88.9        | 78.4        | 75.1          | 87.9         |
| Winnow-12B             | Gemma 4 12B | ✓       | 65.9          | 85.7        | 75.8        | 71.6          | 85.3         |
| Cygnnet                | Gemma 4 12B | –       | 63.8          | 87.3        | 75.6        | 73.8          | 87.9         |
| Clef                   | Qwen3.8-27B | ✓       | 65.4          | 83.4        | 74.4        | 75.1          | 87.4         |
| Plumb-4B               | Qwen3.5-4B  | ✓       | 60.4          | 85.8        | 73.1        | 75.3          | 89.2         |
| Open-Jev-27B           | Qwen3.8-27B | ✓       | 59.0          | 83.7        | 71.3        | 67.2          | 85.3         |
| decider-4b-v2          | Qwen3.5-4B  | ✓       | 56.1          | 85.5        | 70.8        | 69.6          | 83.5         |
| SemIf                  | Qwen3.5-4B  | –       | 46.3          | 79.0        | 62.7        | 62.4          | 80.1         |
| Quyet-1.0-Medium       | Qwen3.5-4B  | ?       | 51.0          | 66.9        | 58.9        | 59.4          | 81.4         |

**Table 4. Performance on seven held-out public decision datasets.** Accuracy (%) for all models evaluated on the same questions. TREC (Li & Roth, 2002): question classification; CB (de Marneffe et al., 2019): natural language inference; WANLI (Liu et al., 2022): natural language inference; Auth. and Pert.: SemIf’s authored and perturbation sets (SemIf contributors, 2026); TypeSafe: public decision examples, measured by agreement with the most likely reference answer; decider (Marosi, 2026): 29 held-out classification tasks, averaged across tasks. Macro: average accuracy across the seven datasets. ECE: mean expected calibration error across the six datasets with hard labels (excluding TypeSafe); lower is better. Our models use fitted temperatures; trained results use seed 1. Bold indicates the best result in each column.

| System                                | TREC        | CB          | WANLI       | Auth.       | Pert.        | TypeSafe    | decider     | Macro       | ECE         |
|---------------------------------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------|
| Items                                 | 500         | 56          | 256         | 144         | 108          | 348         | 7,800       | 9,212       |             |
| Jev 1.13.0 (API)                      | 92.4        | 82.1        | 76.6        | 97.2        | <b>100.0</b> | 87.0        | 80.6        | 88.0        | .066        |
| Gemma 4 31B, zero-shot                | 92.8        | 85.7        | <b>78.9</b> | 95.1        | 97.2         | 87.5        | 81.9        | 88.4        | .077        |
| Jeff-1.0-Large (Gemma 4 31B, trained) | 92.8        | 87.5        | 78.5        | 95.8        | <b>100.0</b> | 85.8        | <b>82.8</b> | 89.0        | <b>.047</b> |
| Qwen3.6-27B, zero-shot                | 93.0        | 83.9        | 77.7        | <b>97.9</b> | <b>100.0</b> | 84.3        | 81.4        | 88.3        | .052        |
| Qwen3.6-27B, trained                  | <b>93.4</b> | 87.5        | 77.0        | 97.2        | <b>100.0</b> | <b>88.3</b> | 82.2        | <b>89.4</b> | .064        |
| Gemma 4 12B, zero-shot                | 92.2        | <b>92.9</b> | 71.5        | 95.1        | 99.1         | 84.9        | 78.9        | 87.8        | .080        |
| Gemma 4 12B, trained                  | 91.4        | 87.5        | 73.4        | 93.8        | <b>100.0</b> | 87.2        | 79.8        | 87.6        | .070        |
| Qwen3.5-4B, zero-shot                 | 87.4        | 73.2        | 63.7        | 80.6        | 77.8         | 80.7        | 74.1        | 76.8        | .094        |
| Qwen3.5-4B, trained                   | 92.0        | 83.9        | 73.4        | 90.3        | 94.4         | 81.8        | 78.9        | 85.0        | .080        |

**Knowledge-heavy multiple choice.** On six standard benchmarks (BoolQ, MMLU, MMLU-Pro, ARC-Challenge, WinoGrande, SciQ; Clark et al., 2019; Hendrycks et al., 2021a; Wang et al., 2024; Clark et al., 2018; Sakaguchi et al., 2020; Welbl et al., 2017) Jev is ahead: 92.9 macro accuracy against 88.8 for our trained 31B model, a gap of  $4.0 \pm 0.8$ , mostly on MMLU-Pro, MMLU and WinoGrande

(Appendix D). These tasks test knowledge more than reading a given state. Our trained model leads the strongest public systems (Quyet-1.0-Large 88.3, deck31b 88.1), and training raises accuracy for all four of our models and halves the 31B model’s calibration error (ECE 0.111 to 0.051).

**Table 5. What matters** (Gemma 4 31B, OpenDecisions Capability; 400 steps and seed 1 unless noted; 800, 1,500 and 3,000 steps: mean of seeds 2 and 3).  $\Delta$ : difference to the recipe’s three-seed mean (84.4),  $\pm 1.96$  standard errors (question sampling and seed variation). Pretrained head: trained first on frozen features.

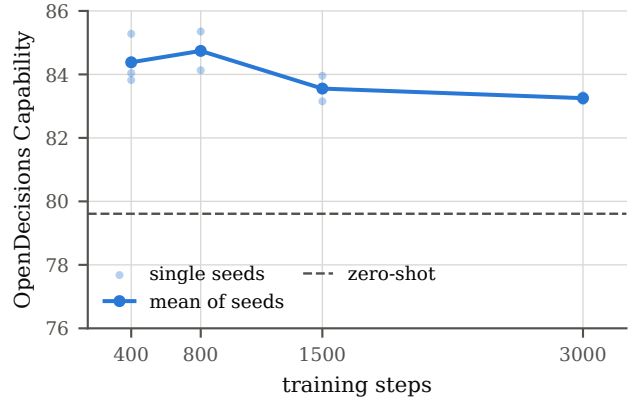
| Variant                                 | Capability | $\Delta$        |
|-----------------------------------------|------------|-----------------|
| Recipe (mean of 3 seeds)                | 84.4       |                 |
| No training (zero-shot)                 | 79.6       | $-4.8 \pm 2.0$  |
| <i>What is trained</i>                  |            |                 |
| Head only, frozen backbone              | 78.9       | $-5.5 \pm 2.5$  |
| LoRA, head frozen at its initialization | 84.9       | $+0.6 \pm 2.1$  |
| <i>Backbone</i>                         |            |                 |
| Gemma base model + LoRA recipe          | 70.3       | $-14.1 \pm 3.3$ |
| <i>Head initialization</i>              |            |                 |
| LoRA, pretrained head, trained jointly  | 82.5       | $-1.9 \pm 2.4$  |
| LoRA, pretrained head kept frozen       | 81.6       | $-2.8 \pm 2.4$  |
| <i>LoRA rank</i>                        |            |                 |
| Rank 16                                 | 84.7       | $+0.3 \pm 2.1$  |
| Rank 64                                 | 84.5       | $+0.1 \pm 2.2$  |
| Rank 128                                | 84.9       | $+0.6 \pm 2.2$  |
| <i>Training data</i>                    |            |                 |
| Public data only (no teacher labels)    | 82.5       | $-1.9 \pm 2.2$  |
| Hard labels instead of soft targets     | 84.1       | $-0.2 \pm 2.1$  |
| <i>Training length (with decay)</i>     |            |                 |
| 800 steps                               | 84.7       | $+0.4 \pm 1.7$  |
| 1,200 steps                             | 83.9       | $-0.4 \pm 2.2$  |
| 1,500 steps                             | 83.6       | $-0.8 \pm 1.8$  |
| 2,300 steps                             | 82.9       | $-1.4 \pm 2.2$  |
| 3,000 steps                             | 83.3       | $-1.1 \pm 1.9$  |
| 3,900 steps                             | 82.6       | $-1.8 \pm 2.3$  |

#### 5.4. Ablations

We examine which parts of the training recipe improve performance and which can be simplified. Table 5 varies one component of the Gemma 4 31B recipe at a time, using its three-seed mean of 84.4 Capability as the reference. Most variants use a single run with seed 1; the training-length experiments also include seeds 2 and 3. Reported intervals account for question sampling and estimated training-seed variability, so small differences should be interpreted cautiously.

**LoRA drives the improvement.** Training only a decision head on frozen model representations gives 78.9 Capability, no improvement over the calibrated zero-shot score of 79.6. In contrast, training LoRA while keeping the head fixed at its original answer-letter weights gives 84.9, comparable to the 84.4 reference score from joint training. The main benefit therefore comes from adapting the language model, with no clear additional gain from training the head.

**Preserving the original head initialization.** We also test whether training the head before LoRA training provides a better starting point. We first train a head on frozen repre-



**Figure 3. Training length for Gemma 4 31B.** Capability on OpenDecisions (Table 9): mean over seeds (line) and single seeds (dots). At 400 steps: the three seeds of Table 3; at 800 to 3,000 steps: seeds 2 and 3, each after a final learning-rate decay branched from one longer run per seed. Each step processes 65,536 padded tokens. Dashed: calibrated zero-shot baseline. Smaller-model results appear in Figure 4 (Appendix B).

sentations, then use it to initialize LoRA fine-tuning, either updating the head jointly or keeping it fixed. These variants score 1.9–2.8 points below the reference. Starting directly from the answer-letter slice of the pretrained LLM head avoids this additional training stage and preserves zero-shot performance at LoRA initialization.

**Starting from an instruction-tuned model.** Our short fine-tuning procedure does not replace the benefits of an instruction-tuned starting checkpoint. Applied to pretrained-only Gemma 4 31B, the recipe reaches 70.3 Capability: 14.1 points below the instruction-tuned model after fine-tuning, and 9.3 points below that model before fine-tuning (see Appendix H for more details).

**Longer training offers no consistent benefit.** All three Gemma 4 31B seeds achieve their highest observed scores at 400 or 800 steps and score lower at later checkpoints (Figure 3 and Table 5). The decline is mainly in Calibration. The smaller models show a similar pattern under their tested recipes (Figure 4).

**Stable across LoRA ranks.** LoRA ranks 16, 32, 64, and 128 give almost identical Capability scores, indicating that a smaller LoRA rank could potentially make training cheaper without meaningful loss in performance; however, these results are only one seed each.

**Teacher labels help, soft labels (possibly) don’t.** Removing the approximately 50,000 questions labeled by language models gives 82.5, a decrease of  $1.9 \pm 2.2$  points. Replacing soft targets with hard labels (via argmax) produces no measurable change in Capability.

**Alternative encodings and training objectives.** We also test an option encoding designed to make predictions independent of option order, along with alternative losses for the decision head (Appendix F). The order-invariant encoding provides no clear benefit on Qwen3.5-4B. When training a head on frozen representations, none of the tested losses clearly improves calibration after temperature scaling. An expected-utility objective, which computes the expected reward for correct, incorrect, and abstained answers, improves accuracy by  $0.9 \pm 0.9$  percentage points over cross-entropy. Combined with LoRA on Gemma 4 31B, the expected-utility objective stays within the range of the recipe’s three seeds (Appendix F).

## 6. Discussion and limitations

**Building on existing capabilities.** Our results show that open-weight language models already achieve performance comparable to Jev *without any additional training* and without the need for specialized architectures or calibrated reinforcement learning. We show that temperature scaling further improves the calibration of these models. These findings reveal the latent capabilities of open-source LLMs for probabilistic classification.

**Simple LoRA fine-tuning goes beyond Jev.** We show that simple LoRA fine-tuning can improve the model’s decision capabilities even further. Our ablations suggest that the main benefit comes from LoRA itself rather than learning a new decision head. We show LoRA fine-tuning with the original head fixed performs comparably to joint training. This suggests that the simple answer-letter readout is already effective, and that improving the representations it reads is more important in our setting. LoRA with a fixed readout therefore provides a useful baseline for testing more specialized decision-model designs.

**Data quality and limitations.** We open-source OpenDecisions, a validation dataset for decision models. Although OpenDecisions follows JevBench’s published task composition, it may not represent all real-world applications. The dataset is 78% English, and many questions were generated by language models rather than written by real users. Qwen3.6-27B provides part of the training supervision, while the related Qwen3.8-27B participates in evaluation labeling, which may introduce bias. Finally, our implementation supports at most 16 answer options.

**Future work.** Although our findings reveal the latent capabilities of open-source models for probabilistic classification, this does not establish that reinforcement learning or specialized decision architectures cannot improve performance further. In fact, we believe that exploring novel architectures, training procedures, and RL methods to fur-

ther improve performance can be an interesting avenue for future work. Another avenue of interest is exploring which training examples actually improve decision-making; while training data for LLMs is abundant, more high-quality data tailored towards decision models will likely help improve these models further.

## Acknowledgments

JG is supported by a Kempner Graduate Fellowship. DAM acknowledges support from the Kempner Institute and NSF award no. 2229881.

## AI Statement

The authors used AI agents to help with writing code, running experiments, and writing. AI-generated content was thoroughly verified, and the authors take responsibility for the final contents of this work.

## References

- allebee. JevK5. <https://github.com/allebee/jevK5>, 2026.
- Almeida, D. Introducing system one models & Jev. Type-Safe AI blog, <https://typesafe.ai/blog/introducing-system-one-models-and-jev>, 2026.
- Anthropic. Claude Opus model family. <https://www.anthropic.com/claude/opus>, 2026.
- blockbrain-ai. Cygnet recipe. <https://github.com/blockbrain-ai/cygnet-recipe>, 2026.
- Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of the 2019 World Wide Web Conference*, 2019. URL <https://dl.acm.org/doi/10.1145/3308560.3317593>.
- Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. URL [https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493\\_1950\\_078\\_0001\\_vofeit\\_2\\_0\\_co\\_2.xml](https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml).
- Cai, Z. Open-Jev-27B v1.1. <https://huggingface.co/ZefanCai/Open-Jev-27B-v1.1>, 2026.
- Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In

- Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017. URL <https://aclanthology.org/S17-2001/>.
- Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., and Stoica, I. Chatbot arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning (ICML)*, 2024. URL <https://proceedings.mlr.press/v235/chiang24b.html>.
- Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, 2019. URL <https://aclanthology.org/N19-1300/>.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL <https://arxiv.org/abs/1803.05457>.
- Cloudflare. Clef. <https://huggingface.co/Cloudflare/clef>, 2026.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. Free Dolly: Introducing the world’s first truly open instruction-tuned LLM. Databricks blog, 2023. URL <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>.
- crh225. Plumb-4B. <https://github.com/crh225/plumb>, 2026.
- de Marneffe, M.-C., Simons, M., and Tonhauser, J. The CommitmentBank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23*, 2019. URL <https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601>.
- denis-pplx. AutoJev-27B. <https://huggingface.co/denis-pplx/autojev-27b>, 2026.
- Desai, S. and Durrett, G. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL <https://aclanthology.org/2020.emnlp-main.21/>.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423/>.
- EldanRing. Winnov-12B. <https://huggingface.co/EldanRing/Winnov-12B>, 2026.
- Epstein, E. S. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8(6): 985–987, 1969. URL [https://journals.ametsoc.org/view/journals/apme/8/6/1520-0450\\_1969\\_008\\_0985\\_assfpf\\_2\\_0\\_co\\_2.xml](https://journals.ametsoc.org/view/journals/apme/8/6/1520-0450_1969_008_0985_assfpf_2_0_co_2.xml).
- Gemma Team, Google DeepMind. Gemma 4 model card (gemma-4-31b-it, gemma-4-12b-it). <https://huggingface.co/google/gemma-4-31b-it>, 2026. Accessed October 2026.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017. URL <https://proceedings.mlr.press/v70/guo17a.html>.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL <https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html>.
- HopitAI. Hopper. <https://huggingface.co/HopitAI/hopper>, 2026.
- Howard, J. and Ruder, S. Universal language model finetuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339. Association for Computational Linguistics, 2018. doi:

- 10.18653/v1/P18-1031. URL <https://aclanthology.org/P18-1031/>.
- IkerMoel. open-alternative-jev. <https://github.com/ikermoel/open-alternative-jev>, 2026.
- JevBench maintainers. JevBench: a benchmark for Jev-type decision models. <https://benchmarkheaven.com/jev-models>; harness: <https://github.com/fstandhartinger/jevbench>, 2026. The paper reports results for versions 1.6.x and 1.7; accessed October 10, 2026.
- Kadavath, S. et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL <https://arxiv.org/abs/2207.05221>.
- Karandikar, A., Cain, N., Tran, D., Lakshminarayanan, B., Shlens, J., Mozer, M. C., and Roelofs, B. Soft calibration objectives for neural networks. In *Advances in Neural Information Processing Systems*, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/f8905bd3df64ace64a68e154ba72f24c-Abstract.html>.
- Kennedy, C. J., Bacon, G., Sahn, A., and von Vacano, C. Constructing interval variables via faceted Rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*, 2020. URL <https://arxiv.org/abs/2009.10277v1>. Version 1, September 22, 2020.
- krishna-gogineni-765. deck31b. <https://github.com/krishna-gogineni-765/deck31b>, 2026.
- kshetrajna12. reflex. <https://github.com/kshetrajna12/reflex>, 2026.
- Kumar, A., Sarawagi, S., and Jain, U. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL <https://proceedings.mlr.press/v80/kumar18a.html>.
- Li, X. and Roth, D. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, 2002. URL <https://aclanthology.org/C02-1150/>.
- Li, Y. and Wagle, J. LLM-as-Jev: LLMs are already Jev-style decision models – when and how to fine-tune them. *arXiv preprint arXiv:2610.02076*, 2026. URL <https://arxiv.org/abs/2610.02076v2>. Version 2.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL <https://arxiv.org/abs/2305.20050>.
- Liu, A., Swayamdipta, S., Smith, N. A., and Choi, Y. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022. URL <https://aclanthology.org/2022.findings-emnlp.508/>.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Marosi, M. decider: one-pass typed decisions with calibrated probabilities. <https://github.com/Mapika/decider>, 2026. Apache-2.0; accessed October 2026.
- Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P. H. S., and Dokania, P. K. Calibrating deep neural networks using focal loss. In *Advances in Neural Information Processing Systems*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/aeb7b30ef1d024a76f21a1d40e30c302-Abstract.html>.
- Nguyen, C. Quyet 1.0: calibrated decision models. <https://huggingface.co/chinhnc/Quyet-1.0-Large>, 2026. Model family includes Large, Medium, Small, and Tiny variants; accessed October 10, 2026.
- Nie, Y., Zhou, X., and Bansal, M. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL <https://aclanthology.org/2020.emnlp-main.734/>.
- OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL <https://arxiv.org/abs/2508.10925>.
- Pakdaman Naeini, M., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI Conference on Artificial Intelligence*, 2015. URL <https://ojs.aaai.org/index.php/AAAI/article/view/9602>.
- Pezeshkpour, P. and Hruschka, E. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023. URL <https://arxiv.org/abs/2308.11483>.
- Qwen Team. Qwen3.5-4B model card. <https://huggingface.co/Qwen/Qwen3.5-4B>, 2026a.
- Qwen Team. Qwen3.6-27B model card. <https://huggingface.co/Qwen/Qwen3.6-27B>, 2026b.

- Qwen Team. Qwen3.8-27B model card. <https://huggingface.co/Qwen/Qwen3.8-27B>, 2026c.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. OpenAI technical report, 2018. URL [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf).
- Robinson, J. and Wingate, D. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations (ICLR)*, 2023. URL <https://openreview.net/forum?id=yKbprarjc5B>.
- Sakaguchi, K., Le Bras, R., Bhagavatula, C., and Choi, Y. WinoGrande: An adversarial Winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6399>.
- Schick, T. and Schütze, H. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.20. URL <https://aclanthology.org/2021.eacl-main.20/>.
- SemIf contributors. SemIf: zero-shot decision scoring with letter-token logits. <https://github.com/TheoLeeCJ/SemIf-OpenJev>, 2026. MIT license; accessed October 2026.
- Stepanov, I., Shtopko, M., Vodianytskyi, D., Lukashov, O., Yavorskyi, A., and Yaroshenko, M. GLiClass: Generalist lightweight model for sequence classification tasks. *arXiv preprint arXiv:2508.07662*, 2025. URL <https://arxiv.org/abs/2508.07662>.
- Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. D. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL <https://aclanthology.org/2023.emnlp-main.330/>.
- torchcast-ai. Torchcast-decision-12B. <https://huggingface.co/torchcast-ai/torchcast-decision-12b>, 2026.
- Vicentino, C. Eikos: open, calibrated, single-pass typed-decision models for finance and trading. <https://huggingface.co/caiovicentino1/Eikos-27B>, 2026.
- Wang, Y. et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL [https://proceedings.neurips.cc/paper\\_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Dataset\\_s\\_and\\_Benchmarks\\_Track.html](https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Dataset_s_and_Benchmarks_Track.html).
- Welbl, J., Liu, N. F., and Gardner, M. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 2017. URL <https://aclanthology.org/W17-4413/>.
- Yin, W., Hay, J., and Roth, D. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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. 3914–3923. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1404. URL <https://aclanthology.org/D19-1404/>.
- Yinsongxu. LLM2Jev. <https://github.com/Yinsongxu/LLM2Jev>, 2026.
- Z.ai. GLM-4.7-Flash model card. <https://huggingface.co/zai-org/GLM-4.7-Flash>, 2026.
- Zhang, J., Yang, T., Shi, Y., and Wu, L. AnyJev: Turn any LLM into a Jev-style decision model. <https://github.com/nokia-applied-research/AnyJev>, 2026.
- Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=shr9PXz7T0>.

## A. Request format

A request has a state (text or a JSON object) and one or more named questions in TypeSafe’s format, as JevBench sends them. The example below has one question of each type (line breaks added for display). Each question is answered in its own forward pass.

```
{
  "state": "Order #4411 was charged twice. The second charge is still pending, and the customer asks for the money back.",
  "questions": {
    "team": {
      "type": "choice", "instructions": "Which team should handle this ticket?",
      "criteria": {
        "billing": "Payments, charges and invoices.",
        "technical": "Bugs, errors and login problems.",
        "refund": "Returns of purchased items."
      }
    },
    "money_owed": {
      "type": "noul", "instructions": "Is the customer owed money back?",
      "criteria": {
        "true": "The customer was charged more than they should pay.",
        "false": "Every charge is correct."
      }
    },
    "urgency": {
      "type": "score", "instructions": "How urgent is this ticket?",
      "criteria": [
        "Not urgent: can wait a week.",
        "Normal: handle within two days.",
        "Urgent: handle today.",
        "Critical: handle now."
      ]
    }
  }
}
```

**System message.** The same for every question and every model (the prompt of SemIf ([SemIf contributors, 2026](#))):

```
Apply the supplied criterion to the supplied evidence. Choose exactly one listed option. Respond with only its uppercase letter, with no explanation or reasoning.
```

**User message.** One JSON object: the state (evidence), the question’s instructions (criterion) and the options, labeled A, B, C, and so on, each described as “<id>: <text>”. A *choice question* lists its options in the order of the request. A *yes/no question* becomes two options, true and false. A *score question* lists its levels from lowest to highest, with ids 0, 1, 2, and so on. The three user messages of the example (line breaks added for display; each message is one line):

choice

```
{
  "evidence": "Order #4411 was charged twice. The second charge is still pending, and the customer asks for the money back.",
  "criterion": "Which team should handle this ticket?",
  "options": [
    { "letter": "A", "description": "billing: Payments, charges and invoices." },
    { "letter": "B", "description": "technical: Bugs, errors and login problems." },
    { "letter": "C", "description": "refund: Returns of purchased items." }
  ]
}
```

yes/no

```
{
  "evidence": "Order #4411 was charged twice. The second charge is still pending, and the customer asks for the money back.",
  "criterion": "Is the customer owed money back?",
  "options": [
    { "letter": "A", "description": "true: The customer was charged more than they should pay." },
    { "letter": "B", "description": "false: Every charge is correct." }
  ]
}
```

score

```
{
  "evidence": "Order #4411 was charged twice. The second charge is still pending, and the customer asks for the money back.",
  "criterion": "How urgent is this ticket?",
  "options": [
    { "letter": "A", "description": "0: Not urgent: can wait a week." },
    { "letter": "B", "description": "1: Normal: handle within two days." },
    { "letter": "C", "description": "2: Urgent: handle today." },
    { "letter": "D", "description": "3: Critical: handle now." }
  ]
}
```

**Complete model input.** The two messages are rendered with the model’s own chat template, with thinking disabled: the template opens the model’s turn and closes an empty thinking block, so the next token is the answer. For Gemma 4 the choice question becomes this string of 145 tokens (the user message as above):

Gemma 4

```
<bos><|turn>system
Apply the supplied criterion to the supplied evidence. Choose exactly one listed option. Respond with only its uppercase letter, with no explanation or reasoning.<turn|>
<|turn>user
{"evidence": "Order #4411 was charged twice. ...", "criterion": "Which team should handle this ticket?", "options": [...]}<turn|>
<|turn>model
<|channel>thought
<channel|>
```

For the Qwen models (149 tokens; the string ends with two line breaks after </think>):

Qwen3.6 and Qwen3.5

```
<|im_start|>system
Apply the supplied criterion to the supplied evidence. Choose exactly one listed option. Respond with only
its uppercase letter, with no explanation or reasoning.<|im_end|>
<|im_start|>user
{"evidence": "Order #4411 was charged twice. ...", "criterion": "Which team should handle this ticket?",
 "options": [...]}<|im_end|>
<|im_start|>assistant
<think>

</think>
```

**Output.** The model reads the string once. The decision head (Eq. 2) maps the hidden state at the last position to one logit per option slot; the first  $K$  slots are used, and a softmax over them gives the raw probabilities. Next, we divide the logits by the temperature of the question type and return one answer per question. Table 6 shows the numbers of Jeff-1.0-Large for the example (temperatures 1.52 for choice and 1.31 for yes/no and score questions); the answer it returns is (rounded for display):

```
{
  "team": {
    "type": "choice", "choice": "billing",
    "probabilities": {"billing": 0.842, "technical": 0.010, "refund": 0.148}},
  "money_owed": {
    "type": "noul", "noul": 0.992},
  "urgency": {
    "type": "score", "score": 2,
    "probabilities": {"0": 0.020, "1": 0.226, "2": 0.556, "3": 0.198}}
}
```

A yes/no answer is the probability of `true`.

**Table 6. Output of Jeff-1.0-Large for the example request.** Head logit: the decision head’s logit for the option (Gemma’s logits are soft-capped at 30). Raw: softmax over the question’s options. Served: after the temperature.

| Question            | Letter | Option        | Head logit | Raw   | Served       |
|---------------------|--------|---------------|------------|-------|--------------|
| team (choice)       | A      | billing       | 26.41      | 0.933 | <b>0.842</b> |
|                     | B      | technical     | 19.65      | 0.001 | 0.010        |
|                     | C      | refund        | 23.77      | 0.066 | 0.148        |
| money_owed (yes/no) | A      | true          | 24.13      | 0.998 | <b>0.992</b> |
|                     | B      | false         | 17.74      | 0.002 | 0.008        |
| urgency (score)     | A      | 0: not urgent | 23.03      | 0.008 | 0.020        |
|                     | B      | 1: normal     | 26.22      | 0.194 | 0.226        |
|                     | C      | 2: urgent     | 27.40      | 0.634 | <b>0.556</b> |
|                     | D      | 3: critical   | 26.05      | 0.164 | 0.198        |

**Training questions.** Training uses the same system message, user message and chat template, with two differences in how the options of a training question are shown. A training question comes from a dataset (Table 2) and keeps that dataset’s option ids and wording; a yes/no question keeps its own two options (for example “yes, entailment” and “no, not entailment”). In each epoch the options are shown in a new random order; score levels keep their order. For Qwen3.5-4B, a fixed half of the training questions is instead shown in the evaluation format above: a yes/no question becomes the options `true` and `false`, `true` first, and a choice question lists its options in one fixed order (sorted by id, since a training question has no request order). The other models are trained on the first format only. The options of one yes/no training question in both formats (texts shortened):

```
training format (one epoch)
"options": [{"letter": "A", "description": "no, not entailment: Text 1 does not support Text 2 ..."},
 {"letter": "B", "description": "yes, entailment: Text 1 logically guarantees Text 2 ..."}]
```

```
evaluation format
"options": [{"letter": "A", "description": "true: entailment: Text 1 logically guarantees Text 2 ..."},
 {"letter": "B", "description": "false: not entailment: Text 1 does not support Text 2 ..."}]
```

Table 7. Training settings of the recipe. Qwen3.5-4B differs in three settings (last row).

| Setting     | Value                                                                                                               |
|-------------|---------------------------------------------------------------------------------------------------------------------|
| LoRA        | rank 32, $\alpha = 64$ , dropout 0; q/k/v/o, gate/up/down projections (Qwen: also in_proj_qkv, in_proj_z, out_proj) |
| Head        | decision head (Eq. 2), 16 slots, initialized from the language-model head, trained jointly                          |
| Optimizer   | AdamW, LR $2 \cdot 10^{-5}$ (LoRA), $10^{-4}$ (head), weight decay 0                                                |
| Schedule    | 250 warmup steps, constant to step 300, cosine decay to 0 at step 400 (WSD)                                         |
| Batch       | 65,536 padded tokens per step: 4 GPUs $\times$ 8,192 tokens $\times$ 2 accumulation steps                           |
| Inputs      | at most 8,192 tokens (longer rows skipped); options shuffled; bf16; gradient checkpointing                          |
| Loss        | cross-entropy against soft targets where available, else the label; score questions included                        |
| Checkpoint  | the final step (no early stopping)                                                                                  |
| Calibration | per-type temperatures fitted by NLL on our validation set (3,884 rows)                                              |
| Seeds       | 1, 2, 3 for every model                                                                                             |
| Hardware    | 4 $\times$ H200 or H100; 400 steps take 78 min (31B), 64 min (27B), about 40 min (12B), 23 min (4B)                 |
| Qwen3.5-4B  | LoRA LR $10^{-4}$ ; distillation from the trained 31B model; half of the questions in evaluation format             |

## B. Training details

**WSD branches.** For the longer runs (Figures 3 and 4, Table 5) we train one run at a constant learning rate and branch a 100- to 400-step cosine decay off saved states at steps 300, 1,000, 2,000 and 3,500 (31B, seed 1). For seeds 2 and 3 of the 31B model and for the smaller models we branch 100-step decays at steps 300, 700 and 1,400, and the run itself decays from step 2,900 to 3,000 (Qwen3.6-27B and seed 1 of Gemma 4 12B: from 1,400 to 1,500). A branch starts from the full saved training state (weights, optimizer and data position), so it shares the main run’s first steps exactly. The 400-step recipe is the branch at step 300; a unit test checks that a branch equals a direct run with the same schedule.

**Qwen3.5-4B.** The 31B recipe does not improve Qwen3.5-4B (Table 9). We tried four single changes at 400 steps (two seeds each) and combined three of them: (i) a LoRA learning rate of  $10^{-4}$  instead of  $2 \cdot 10^{-5}$  (63.4 alone, against 63.0 for the recipe); (ii) distillation: each target is the average of the pool’s target and the prediction of our trained Gemma 4 31B (seed 1, with its fitted temperatures; 65.1 alone); (iii) a fixed half of the training questions is shown in the format of the evaluation questions: options in a fixed order instead of shuffled, and yes/no questions with the options true and false, in this order (63.8 alone). The fourth change, averaging each target with the zero-shot prediction, scored 64.0 and was not used. Together, the three changes give 66.6. We chose them on OpenDecisions, so this number is optimistic in the same way as the 31B number.

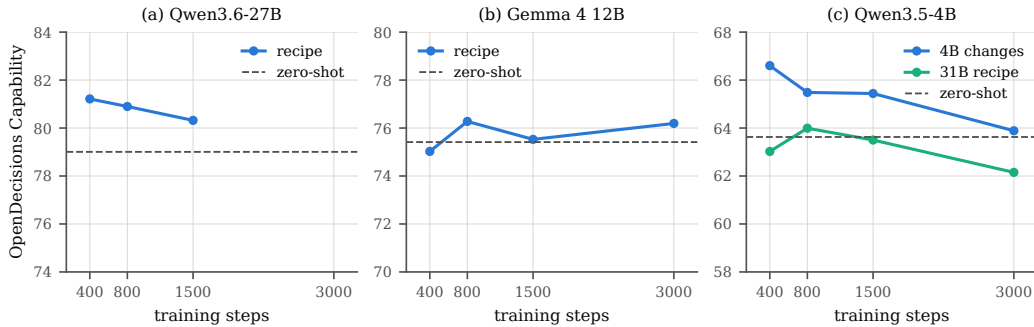


Figure 4. **Training length for the smaller models.** OpenDecisions Capability against training steps (65,536 tokens each), for checkpoints with a learning-rate decay at the end; each point is the mean of three seeds (Table 9; Gemma 4 12B at 3,000 steps: seeds 2 and 3). (a) Qwen3.6-27B, recipe. (b) Gemma 4 12B, recipe. (c) Qwen3.5-4B: with the three changes for this model (Section 4) and with the 31B recipe. Dashed: zero-shot.

## C. OpenDecisions details

**Agreement with the public leaderboard.** We ran 17 public systems, including the Jev API, as their authors serve them and compared their scores, under the leaderboard’s own scorer, with their official v1.6.1 Capability (Table 11). For the 10 systems with official Capability of at least 55 that were also used while building OpenDecisions, OpenDecisions ranks them in nearly the official order (Spearman 0.98). Four systems were never used while building the set (Eikos-27B, Clef,

Table 8. **Per-seed OpenDecisions Capability** of the trained models in Table 3.

| Model                   | Steps | Zero-shot | Seed 1 | Seed 2 | Seed 3 | Mean |
|-------------------------|-------|-----------|--------|--------|--------|------|
| Gemma 4 31B             | 400   | 79.6      | 85.3   | 84.0   | 83.8   | 84.4 |
| Qwen3.6-27B             | 400   | 79.0      | 81.8   | 81.7   | 80.2   | 81.2 |
| Gemma 4 12B             | 400   | 75.4      | 75.6   | 74.6   | 74.8   | 75.0 |
| Qwen3.5-4B (4B changes) | 400   | 63.6      | 68.0   | 66.4   | 65.5   | 66.6 |

 Table 9. **Training length** (OpenDecisions Capability, decayed checkpoints, mean of three seeds; Gemma 4 31B at 800 to 3,000 steps and Gemma 4 12B at 3,000 steps: seeds 2 and 3; the numbers of Figures 3 and 4).

| Model and variant      | 400  | 800  | 1,500 | 3,000 |
|------------------------|------|------|-------|-------|
| Gemma 4 31B            | 84.4 | 84.7 | 83.6  | 83.3  |
| Qwen3.6-27B            | 81.2 | 80.9 | 80.3  | —     |
| Gemma 4 12B            | 75.0 | 76.3 | 75.5  | 76.2  |
| Qwen3.5-4B, 4B changes | 66.6 | 65.5 | 65.4  | 63.9  |
| Qwen3.5-4B, 31B recipe | 63.0 | 64.0 | 63.5  | 62.2  |

Table 10. OpenDecisions composition by question type and tier (target in brackets: the composition of the public JevBench v1.6 split times 8).

| Type   | Easy      | Standard  | Judge     | Hard      | Total         |
|--------|-----------|-----------|-----------|-----------|---------------|
| Choice | 261 (264) | 136 (136) | 255 (256) | 526 (544) | 1,178 (1,200) |
| Yes/no | 110 (112) | 144 (144) | 158 (160) | 174 (184) | 586 (600)     |
| Score  | 72 (72)   | 127 (128) | 168 (168) | 221 (232) | 588 (600)     |

AutoJev-27B and Open-Jev-27B; official Capability 68.0 to 77.2). A line fitted on the 10 old systems predicts their official scores with a root mean square error of 3.0 points (95% interval 2.2 to 3.9) and a mean error of +0.5; 78% of new-old pairs come out in the official order. Over all 14 systems the Spearman correlation is 0.90. JevBench-Open predicts worse (error 5.4), and our training-distribution validation set fails (16.2).

 Table 11. **Does a set predict the official leaderboard?** Agreement between Capability on each set and the official v1.6.1 Capability of public systems with official Capability  $\geq 55$ , each run as its authors serve it and scored with the leaderboard’s scorer. LOO = mean absolute error when predicting each system’s official score from the others by a linear fit. Gray brackets: 95% intervals from resampling item groups (correlations are bounded, so we show the interval itself rather than  $\pm$ ).

| Set                                                     | Items | Spearman          | Pairs in order | LOO  |
|---------------------------------------------------------|-------|-------------------|----------------|------|
| <i>10 systems used while building OpenDecisions</i>     |       |                   |                |      |
| JevBench-Open                                           | 231   | 0.85 [0.56, 0.94] | 0.84           | 4.9  |
| OpenDecisions pilot                                     | 283   | 0.90 [0.67, 0.92] | 0.89           | 4.3  |
| OpenDecisions                                           | 2,352 | 0.98 [0.92, 0.99] | 0.96           | 2.6  |
| Training-distribution validation set                    | 3,884 | 0.45 [0.31, 0.53] | 0.67           | 10.9 |
| <i>Adding 4 systems not used to build OpenDecisions</i> |       |                   |                |      |
| OpenDecisions, new systems only                         | 2,352 | 0.80              | 0.83           | 4.1  |
| OpenDecisions, all 14 systems                           | 2,352 | 0.90 [0.88, 0.95] | 0.87           | 2.7  |

**Results per family.** The families of OpenDecisions separate systems in different ways (Table 12). The trained 31B model is best or close to best on most families; Quyet-1.0-Large leads on fairness pairs and composite rules, Eikos-27B on use cases and perturbation pairs, and Jev on counting traps and calibration items. Every system gets the negation pairs right.

### C.1. Training–evaluation overlap and distribution differences

We compare OpenDecisions with the 417,825 questions in our training pool. This analysis uses an earlier draw of OpenDecisions with 2,389 questions, which contains all 2,352 final questions. We examine whether evaluation questions appear in the training data and how similar the two datasets are. The comparisons use the stored questions, supporting context, and answer options, rather than the exact prompts seen by the model.

Table 12. **Competence by item family on OpenDecisions** (chance-corrected accuracy; same items for all systems; ours with temperatures, trained = mean of three seeds; public systems as in Table 3).

| Family                  | Gemma 4 31B |           | Qwen3.6-27B |           | Eikos-27B   | Quyet-L     | deck31b | Jev         | Torchcast |
|-------------------------|-------------|-----------|-------------|-----------|-------------|-------------|---------|-------------|-----------|
|                         | trained     | zero-shot | trained     | zero-shot |             |             |         |             |           |
| LLM routing             | 78.3        | 76.1      | 75.0        | 72.9      | 76.6        | 77.7        | 76.5    | <b>79.3</b> | 74.3      |
| Multilingual            | <b>80.8</b> | 76.8      | 75.0        | 66.4      | 78.4        | 76.6        | 78.0    | 67.5        | 69.2      |
| Fairness pairs          | 78.8        | 75.4      | 72.4        | 61.8      | 80.5        | <b>86.6</b> | 76.9    | 72.0        | 69.7      |
| Use cases               | 73.4        | 68.6      | 65.8        | 50.7      | <b>74.4</b> | 71.6        | 68.1    | 54.4        | 59.4      |
| Counting traps          | 56.7        | 53.4      | 48.0        | 41.7      | 44.8        | 57.4        | 56.8    | <b>60.4</b> | 41.3      |
| Negation pairs          | 100         | 100       | 100         | 100       | 100         | 100         | 100     | 100         | 98.9      |
| Option order            | <b>87.2</b> | 85.5      | 83.8        | 81.3      | 86.5        | 84.4        | 85.5    | 85.5        | 84.4      |
| Perturbation pairs      | 67.4        | 55.5      | 66.3        | 58.0      | <b>75.3</b> | 68.0        | 63.1    | 68.3        | 63.6      |
| Composite rules         | 80.3        | 79.3      | 77.8        | 61.9      | 76.1        | <b>85.6</b> | 77.1    | 76.0        | 67.8      |
| Calibration (generated) | 82.9        | 65.7      | 83.6        | 70.0      | <b>85.9</b> | 83.1        | 74.9    | 77.3        | 75.8      |
| Calibration             | 84.5        | 57.8      | 87.6        | 59.4      | 84.9        | 83.0        | 73.2    | <b>94.1</b> | 69.6      |

**Exact overlap.** We check for exact matches between training and evaluation using the complete question inputs, question text alone, and supporting context alone. We repeat these checks after standardizing whitespace and Unicode. We find no exact matches in any of the 2,389 evaluation questions. This rules out direct reuse of identical inputs under these checks, but does not rule out paraphrased questions or shared underlying skills.

**Similarity to training data.** We also measure how closely evaluation questions resemble training questions. We hold out 2,389 training questions ( $H$ ) and use the remaining 415,436 ( $R$ ) as a reference set. Using TF-IDF features and cosine similarity, we find the closest question in  $R$  for each question in  $H$  and OpenDecisions. For this comparison, we remove generic prompt wording and shorten the 37 longest evaluation contexts; the exact-match checks above use the full text.

The median similarity is 0.156 for OpenDecisions, compared with 0.763 for held-out training questions (Table 13). Part of the training similarity comes from repeated supporting contexts. After excluding training questions whose context appears exactly in  $R$ , the median falls to 0.509, still substantially above that of OpenDecisions. The difference also persists when comparisons are restricted to the same question type.

Table 13. Median nearest-neighbor TF-IDF cosine similarity to the same training reference set  $R$ . The last row excludes training questions with an identical supporting context in  $R$ .

| Query set                   | Questions | Median similarity |
|-----------------------------|-----------|-------------------|
| OpenDecisions               | 2,389     | 0.156             |
| Held-out training ( $H$ )   | 2,389     | 0.763             |
| $H$ without shared contexts | 1,110     | 0.509             |

**Distinguishing the datasets.** As a separate check, we train a logistic-regression classifier using TF-IDF features to predict whether a question comes from training or evaluation. We use 2,389 questions from each dataset, grouped five-fold cross-validation, and three random seeds. Related examples are kept in the same fold to reduce leakage, and features are fitted only on the training folds. The classifier achieves 98.82% balanced accuracy, with a standard deviation of 0.10 percentage points across seeds, after removing the identified presentation wrappers. This shows that the datasets differ in observable input text, although differences in wording, formatting, and task composition may contribute to the result.

**Interpretation.** We find no exact matches between OpenDecisions and our training pool, and the two datasets differ substantially in their input distributions. However, this does not mean that the evaluation questions require new skills or are necessarily harder. These differences also do not explain why models perform better or worse. We cannot rule out paraphrased overlap or overlap with the backbones’ pretraining data. Our analysis covers the configured training pool rather than the exact examples used in each training run. Finally, because OpenDecisions was used to select the training recipe, it is a development set rather than an independent test set.

Table 14. **Official JevBench leaderboard** (JevBench maintainers, 2026): the systems with the highest Capability, as published in October 2026 (revision v1.6.7). Each system is measured on its own draw of 1,500 items (300 public, 1,200 sealed); draw: the revision in which the system was measured. Jev is listed as a reference and not ranked.

| System                | Backbone    | Draw   | <i>I</i>    | <i>C</i>    | Capability  |
|-----------------------|-------------|--------|-------------|-------------|-------------|
| Quyet-1.0-Large       | Gemma 4 31B | v1.6.0 | <b>73.4</b> | 90.0        | <b>81.7</b> |
| Jeff-1.0-Large (ours) | Gemma 4 31B | v1.6.2 | 71.9        | <b>91.1</b> | 81.5        |
| decisio v0.8.0        | Gemma 4 31B | v1.6.1 | 70.4        | 88.7        | 79.6        |
| deck31b               | Gemma 4 31B | v1.6.0 | 73.0        | 82.2        | 77.6        |
| Jev 1.13.0 (API)      | closed      | v1.6.1 | 63.6        | 90.6        | 77.1        |

## D. Public evaluation sets

**Held-out decision datasets.** TREC (500 items), CB (56) and WANLI (256, SemIf’s sample) are public classification and inference sets; SemIf’s fixtures (authored, 144 items; perturbations, 108) are public decision items with paired variants; TypeSafe’s example set (348 items) has reference answers written by strong LLMs, so we report agreement with the most likely reference answer; decider’s 29 held-out tasks (7,800 items) are built from public datasets that decider keeps out of its training mixture. A 13-word n-gram check against every row of our training pool finds no overlap for the first six sets and 71 of 7,800 decider items sharing template phrases, which we keep. Inputs up to 16,384 tokens are read. Our own synthetic held-out sets are not in Table 4: on 468 questions with exact answer probabilities, Jev scores 88.0, the zero-shot 31B model 49.6 and the trained one 87.2 (Qwen3.6-27B: 91.0 and 98.1); on 3,529 questions about held-out document domains, Jev 87.3, trained 31B 88.9 and trained 27B 90.1.

**Knowledge-heavy multiple choice.** We draw seeded samples of the sizes used by Li & Wagle (2026): BoolQ (500), MMLU (1,200), MMLU-Pro (800, 10 options), ARC-Challenge (600), WinoGrande (500) and SciQ (400), from splits we did not train on. Our training pool contains the training splits of BoolQ, ARC, MMLU (auxiliary\_train) and WinoGrande, so for our trained models these are seen tasks with unseen items; MMLU-Pro and SciQ are not. A 13-gram check drops 110 items (92 BoolQ passages reappear in BoolQ’s training split with other questions).

Table 15. **Knowledge-heavy multiple-choice benchmarks:** accuracy (%) on our seeded samples without overlapping items. Macro: mean of six tasks. ECE: 15 bins, raw probabilities, mean over tasks. For reference, Li & Wagle (2026) report macro accuracies on their own samples of 78.6 (4B, training-free), 78.9 (4B, LoRA), 82.2 (Winnow-12B), 75.4 (SemIf) and 91.5 (GPT-5.6 Sol, text output).

| System                                | BoolQ       | MMLU        | MMLU-Pro    | ARC-C       | WinoGr.     | SciQ        | Macro       | ECE  |
|---------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------|
| Jev 1.13.0 (API)                      | <b>92.6</b> | <b>91.8</b> | <b>84.0</b> | <b>97.8</b> | <b>92.2</b> | 98.7        | <b>92.9</b> | .032 |
| Quyet-1.0-Large                       | 92.2        | 85.3        | 65.9        | 97.5        | 90.0        | <b>99.0</b> | 88.3        | .035 |
| deck31b                               | 92.4        | 85.8        | 68.8        | 97.2        | 86.0        | 98.7        | 88.1        | .077 |
| decider-4b-v2                         | 91.7        | 75.0        | 52.3        | 92.7        | 83.0        | 98.2        | 82.1        | .038 |
| Torchcast-decision-12B                | 90.0        | 79.0        | 52.8        | 94.3        | 75.8        | 98.0        | 81.6        | .049 |
| Winnow-12B                            | 89.0        | 77.3        | 54.0        | 94.7        | 74.5        | 98.0        | 81.3        | .084 |
| Plumb-4B                              | 88.0        | 73.5        | 53.4        | 92.5        | 67.9        | 98.0        | 78.9        | .073 |
| SemIf                                 | 86.3        | 69.2        | 40.5        | 89.3        | 59.9        | 98.0        | 73.9        | .084 |
| Gemma 4 31B, zero-shot                | 92.4        | 85.9        | 70.3        | 96.8        | 85.0        | 98.7        | 88.2        | .111 |
| Jeff-1.0-Large (Gemma 4 31B, trained) | 92.2        | 86.9        | 71.5        | 97.2        | 86.6        | 98.7        | 88.8        | .051 |
| Qwen3.6-27B, zero-shot                | 91.7        | 84.8        | 62.8        | 96.5        | 84.6        | 98.5        | 86.5        | .055 |
| Qwen3.6-27B, trained                  | 90.9        | 86.9        | 70.6        | 97.2        | 87.0        | 98.7        | 88.5        | .027 |
| Gemma 4 12B, zero-shot                | 88.5        | 75.3        | 54.8        | 93.5        | 69.1        | 98.5        | 80.0        | .167 |
| Gemma 4 12B, trained                  | 90.0        | 77.2        | 54.5        | 94.7        | 72.3        | 98.5        | 81.2        | .064 |
| Qwen3.5-4B, zero-shot                 | 89.0        | 69.8        | 41.3        | 90.3        | 63.3        | 98.0        | 75.3        | .079 |
| Qwen3.5-4B, trained                   | 90.4        | 73.6        | 51.0        | 93.0        | 72.7        | 98.5        | 79.9        | .042 |

## E. Zero-shot accuracy and model size

Figure 5 shows the zero-shot letter readout of eight Qwen models and two Gemma models on the seven held-out decision sets of Table 4, scored in the same way (macro accuracy on the same 9,212 items). Temperatures do not change which option is most likely, so nothing is fitted here. Accuracy rises steeply up to 4B parameters and then levels off: Qwen3.5 scores 48.9 at 0.8B, 57.6 at 2B, 76.8 at 4B, 82.2 at 9B and 86.5 at 27B. From about 12B on, the model family matters as

much as the size. Gemma 4 12B (87.8) is above Qwen3.5-27B (86.5), and Gemma 4 31B (88.4), Qwen3.6-27B (88.3) and Jev (88.0) are within half a point of each other. The two mixture-of-experts models, with 35B parameters of which 3B are active per token, score 86.9 (Qwen3.5) and 86.7 (Qwen3.6), close to the dense 27B model at a fraction of its compute per token. For the Qwen models other than Qwen3.6-27B and Qwen3.5-4B we use earlier zero-shot runs with the same prompt and readout; for these two models, the earlier runs and the runs of Table 4 agree to within 0.1 points.

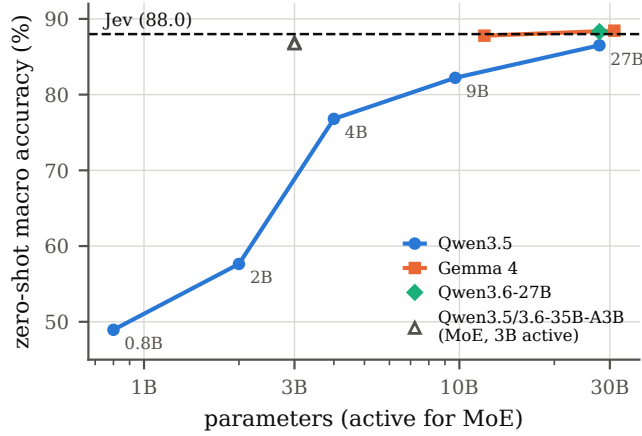


Figure 5. **Zero-shot accuracy against model size:** macro accuracy (%) over the seven held-out decision sets of Table 4, letter readout, no training. Mixture-of-experts models are placed at their 3B active parameters. Dashed: Jev.

## F. Order-invariant encoding and calibration losses

Before we settled on the recipe, we tested two other ideas on Qwen3.5-4B. Both studies use the same training pool (Table 2) but an earlier training setup, so the numbers are not directly comparable to the main text. We rescore all runs on the seven held-out decision sets of Table 4 (Table 16). Intervals include seed variation as in the main text: within each study we pool the seed variance of the variants that have several seeds, and a single run counts it fully.

**Order-invariant encoding.** With the letter readout, the answer can depend on the order of the options. After LoRA training, the 4B model still changes its answer on about 6% of choice and yes/no questions when the option order is reversed. We therefore built an encoding that is order-invariant by construction. The state and question are encoded once. Each option is then appended in its own short segment, all segments get the same positions, and no option can see the others. The score of option  $j$  is read from the final hidden state  $h_j$  of its segment as  $s_j = w_0^\top h_j + a^\top \text{std}(h_j) + b$ , where  $w_0$  is the difference of the language-model-head rows for “yes” and “no”, and  $a, b$  start at zero, as in the decision head (Eq. (2)). The probabilities are the softmax of the scores over the options. We tried three variants: a set layer (two attention layers without positions) that lets the option vectors interact; a clearer wording for each option segment (Is the answer “<option>?”); and the same wording with all options, sorted by text, listed once in the shared part. Each run trains LoRA (rank 16 on all projections, learning rate  $10^{-4}$ ) and the head (learning rate  $10^{-3}$ ) for 600 steps of 262,144 tokens, on the same batches.

All order-invariant variants score below the letter format, by 0.6 to 2.4 points. With one run per variant these differences are within noise, so we can only say that the encoding does not help. The losses are largest on SemIf’s perturbation set and on TypeSafe’s set, where the options must be compared with each other. Listing all options in the shared part removes most of the gap, but then the encoding is close to a letter prompt without the letters. A simpler change gives the same invariance at no cost: sort the options by text before building the letter prompt. The prompt is then the same for every input order of a choice or yes/no question, and accuracy does not change (+1.0 points when we sort only at test time,  $-0.4$  when we also sort in training; both within noise). No variant changes the ECE by more than 0.013. We therefore keep the letter format; sorting the options is a free option when exact invariance is needed.

**Calibration losses.** We trained a head on the frozen 4B model with different losses. The head is a linear map of the final hidden state  $h$ , initialized with the letter rows  $W_0$  (Eq. (1)) and trained on cached hidden states (AdamW, learning rate  $10^{-3}$ , batch 8,192, at most 3,000 steps, early stopping on the validation loss). Let  $p$  be the predicted distribution over the  $K$  options and  $y$  the correct option. Besides cross-entropy  $-\log p_y$ , we tried the Brier score  $\sum_j (p_j - \mathbb{1}[j = y])^2$  (Brier,

1950); focal loss  $-(1 - p_y)^\gamma \log p_y$  with  $\gamma = 2$  (Mukhoti et al., 2020); cross-entropy plus  $\lambda$  times a differentiable ECE with soft bin memberships (Karandikar et al., 2021) or plus  $\lambda$  times MMCE, a kernel measure of calibration error (Kumar et al., 2018); cross-entropy against a target that mixes the label with the zero-shot distribution, with weight  $\alpha$  on the label; and an expected-utility objective. Rows with a soft target use it in place of the label.

**Expected utility.** This objective is our closest analog to TypeSafe’s reinforcement learning for calibrated decisions. The head gets one extra output, “abstain”. Let  $\pi$  be the softmax over the  $K$  options and abstain. A correct answer earns  $+1$ , a wrong answer  $-r$ , and abstaining 0. The loss is the negative expected reward,

$$\mathcal{L}_{\text{EU}} = -\left(\pi_y - r \sum_{j \leq K, j \neq y} \pi_j\right), \quad (3)$$

averaged with cross-entropy on the  $K$  options (weight  $1/2$  each). Policy-gradient reinforcement learning with this reward maximizes the same quantity in expectation; with a known label we compute it exactly and need no sampling. The best policy answers only when its top option has probability at least  $r/(1 + r)$ , so the objective rewards knowing when it is unsure. At test time the abstain output is dropped and the option probabilities are renormalized. We use  $r = 1$  and  $r = 3$ .

**Results.** After temperature scaling, no loss lowers the mean ECE by more than 0.01 compared with cross-entropy (0.092). Focal loss is clearly worse in accuracy ( $-2.1 \pm 1.7$  points), and most other losses change accuracy by less than one point. The expected-utility objective with  $r = 3$  does best: over three seeds it is  $+0.9 \pm 0.9$  points above cross-entropy (79.4 against 78.5). Its raw ECE is lower (0.076 against 0.092), but after temperature scaling it is not (0.094). With the decision-head form of Eq. (2), the same objective gives  $+0.5 \pm 1.4$  points. On the frozen Qwen3.6-27B, no head beats the zero-shot readout in accuracy (88.3), with or without the expected-utility objective. All of these heads stay far below LoRA training: the recipe raises the 4B model to 85.0 on these sets (Table 4). Combined with LoRA on Gemma 4 31B (400 steps, seed 1), the expected-utility objective mixed with cross-entropy scores 83.7 ( $r = 1$ ) and 84.8 ( $r = 3$ ) on OpenDecisions, within the range of the recipe’s three seeds (83.8 to 85.3), and 81.2 without cross-entropy. It is not needed to reach and exceed Jev.

## G. Calibration

Table 17 compares Capability scores before and after temperature scaling, while Table 18 reports the fitted temperatures for each question type. The largest improvements occur for the zero-shot Gemma models, whose raw probabilities are overconfident. Temperature scaling improves Capability by 6.7 points for Gemma 4 31B and 10.0 points for Gemma 4 12B. After fine-tuning, the fitted temperatures are much closer to one, and the improvements are smaller.

For the trained Qwen models, many fitted temperatures are below one, which makes their predictions more confident. Using these temperatures directly reduces Capability from 81.2 to 81.0 for Qwen3.6-27B and from 66.6 to 64.6 for Qwen3.5-4B. This illustrates that temperatures fitted on our separate calibration set do not always transfer well to OpenDecisions. We therefore set any fitted temperature below one to one, preventing calibration from sharpening predictions. This restriction does not affect the Gemma models and is also applied to the public systems.

**Public systems.** We apply the same calibration procedure to all competing systems, fitting a separate temperature for each question type using their predictions on our held-out calibration set. As with our models, we restrict temperatures to be at least one. For each system, we report whichever achieves higher Capability on OpenDecisions: its original predictions or its temperature-scaled predictions (Table 19). Temperature scaling substantially improves several systems, including deck31b (+3.2), SemIf (+3.7), Winnow-12B (+3.8), and Torchcast-decision-12B (+2.8). However, it reduces Jev’s Capability from 80.1 to 78.8, so we report Jev without additional calibration. Six other systems have all fitted temperatures below one and are therefore unchanged by our restriction.

**Where the temperatures act.** Table 20 breaks the effect down for Gemma 4 31B. Temperatures never change the most likely option, so choice and yes/no competence stay the same, and they raise calibration for every question type, most for the zero-shot readout (choice: 58.6 to 84.0). On score questions they also lower competence, because the scorer measures it with the expected level of the predicted distribution, which a temperature above 1 pulls toward the middle (zero-shot: 64.5 to 54.1). Across subsets, the zero-shot readout gains most on authored items (+6.6) and real requests (+6.7), and loses on the negation pairs (100 to 94.3) and on the 35 items with exact answer probabilities. After training, the changes per subset are small, except on these 35 items.

Table 16. **Order-invariant encoding and calibration losses**, rescored on the seven held-out decision sets of Table 4. Acc.: macro accuracy (%).  $\Delta$ : paired difference to the first row of the block. ECE: mean over the six hard-label sets, raw and with one temperature fitted on the run’s own validation predictions (+T). Rows with several seeds show the seed mean.

| Variant                                                       | Seeds | Acc.           | $\Delta$       | ECE  | ECE +T |
|---------------------------------------------------------------|-------|----------------|----------------|------|--------|
| <i>Qwen3.5-4B, option encoding (LoRA and head, 600 steps)</i> |       |                |                |      |        |
| Letters, given order                                          | 2     | $85.4 \pm 2.6$ | –              | .100 | .096   |
| Letters, sorted at test time                                  | 1     | $86.3 \pm 3.3$ | $+1.0 \pm 3.5$ | .103 | .096   |
| Letters, sorted in training and test                          | 2     | $84.9 \pm 2.6$ | $-0.4 \pm 2.9$ | .092 | .089   |
| Option-parallel                                               | 1     | $83.4 \pm 3.6$ | $-1.9 \pm 3.8$ | .092 | .092   |
| + set layer                                                   | 1     | $83.1 \pm 3.6$ | $-2.3 \pm 3.8$ | .102 | .102   |
| clearer wording                                               | 1     | $83.0 \pm 3.6$ | $-2.4 \pm 3.7$ | .085 | .083   |
| clearer wording + sorted option list                          | 1     | $84.8 \pm 3.2$ | $-0.6 \pm 3.6$ | .091 | .087   |
| <i>Qwen3.5-4B, loss for a head on the frozen model</i>        |       |                |                |      |        |
| Cross-entropy                                                 | 3     | $78.5 \pm 2.5$ | –              | .092 | .092   |
| Zero-shot (no training)                                       | 1     | $76.8 \pm 2.8$ | $-1.7 \pm 2.0$ | .109 | .096   |
| Brier score                                                   | 1     | $79.6 \pm 2.6$ | $+1.1 \pm 1.3$ | .093 | .094   |
| Focal loss, $\gamma = 2$                                      | 1     | $76.3 \pm 2.7$ | $-2.1 \pm 1.7$ | .193 | .099   |
| CE + soft-binned ECE, $\lambda = 0.1$                         | 1     | $78.4 \pm 2.6$ | $0.0 \pm 0.6$  | .093 | .095   |
| $\lambda = 0.3$                                               | 3     | $78.5 \pm 2.6$ | $0.0 \pm 0.5$  | .089 | .090   |
| $\lambda = 1$                                                 | 1     | $78.3 \pm 2.7$ | $-0.2 \pm 1.6$ | .102 | .097   |
| $\lambda = 3$                                                 | 1     | $78.4 \pm 2.5$ | $0.0 \pm 1.6$  | .113 | .118   |
| CE + MMCE, $\lambda = 0.5$                                    | 1     | $78.6 \pm 2.6$ | $+0.1 \pm 0.6$ | .090 | .091   |
| $\lambda = 2$                                                 | 1     | $78.8 \pm 2.6$ | $+0.3 \pm 0.6$ | .086 | .086   |
| CE to mixed target, $\alpha = 0.5$                            | 1     | $78.4 \pm 2.7$ | $-0.1 \pm 1.5$ | .083 | .082   |
| $\alpha = 0.8$                                                | 1     | $79.1 \pm 2.5$ | $+0.6 \pm 1.0$ | .088 | .088   |
| Expected utility, $r = 1$                                     | 1     | $79.1 \pm 2.7$ | $+0.6 \pm 1.2$ | .094 | .094   |
| $r = 3$                                                       | 3     | $79.4 \pm 2.6$ | $+0.9 \pm 0.9$ | .076 | .094   |
| <i>Qwen3.6-27B, head on the frozen model</i>                  |       |                |                |      |        |
| Zero-shot (no training)                                       | 1     | $88.3 \pm 1.8$ | –              | .065 | .089   |
| Decision head, cross-entropy                                  | 1     | $87.7 \pm 1.8$ | $-0.6 \pm 0.9$ | .079 | .081   |
| Decision head, mixed target, $\alpha = 0.5$                   | 1     | $88.1 \pm 1.8$ | $-0.2 \pm 0.7$ | .056 | .071   |
| Linear head, expected utility, $r = 3$                        | 1     | $88.2 \pm 1.8$ | $-0.2 \pm 0.9$ | .058 | .063   |

Table 17. **Raw readout vs. fitted temperatures** (OpenDecisions Capability; trained = mean of three seeds; temperatures never below 1).

| Model       | Zero-shot |              | Trained |              |
|-------------|-----------|--------------|---------|--------------|
|             | Raw       | Temperatures | Raw     | Temperatures |
| Gemma 4 31B | 72.9      | 79.6         | 81.9    | 84.4         |
| Qwen3.6-27B | 78.2      | 79.0         | 81.2    | 81.2         |
| Gemma 4 12B | 65.4      | 75.4         | 74.0    | 75.0         |
| Qwen3.5-4B  | 59.0      | 63.6         | 66.6    | 66.6         |

Table 18. **Fitted temperatures** per question type, fitted by NLL on our validation set (3,884 questions). Trained: mean of the three seeds of the models in Table 3,  $\pm 1.96$  standard errors. Italic: below 1; temperatures below 1 are set to 1 when serving (we never sharpen).

| Model       | Version   | Choice          | Yes/no          | Score           |
|-------------|-----------|-----------------|-----------------|-----------------|
| Gemma 4 31B | zero-shot | 5.96            | 7.24            | 7.24            |
|             | trained   | $1.47 \pm 0.09$ | $1.31 \pm 0.07$ | $1.25 \pm 0.07$ |
| Qwen3.6-27B | zero-shot | 1.38            | 1.52            | 1.85            |
|             | trained   | $1.01 \pm 0.03$ | $0.83 \pm 0.03$ | $0.77 \pm 0.04$ |
| Gemma 4 12B | zero-shot | 3.84            | 4.67            | 5.68            |
|             | trained   | $1.19 \pm 0.07$ | $1.06 \pm 0.07$ | $1.10 \pm 0.04$ |
| Qwen3.5-4B  | zero-shot | 1.38            | 1.31            | 1.38            |
|             | trained   | $0.86 \pm 0.03$ | $0.79 \pm 0.07$ | $0.76 \pm 0.02$ |

Table 19. **Public systems with and without fitted temperatures** (OpenDecisions Capability). Table 3 uses the higher of the two. Fitted temperatures before the rule; gray: below 1, so set to 1.

| System                 | As served | With temperatures | Fitted temperatures (choice / yes-no / score) |
|------------------------|-----------|-------------------|-----------------------------------------------|
| Eikos-27B              | 83.1      | 83.1              | 0.61 / 0.69 / 0.79                            |
| Quyet-1.0-Large        | 81.8      | 81.7              | 0.94 / 0.82 / 1.06                            |
| deck31b                | 78.1      | 81.3              | 1.68 / 2.59 / 2.37                            |
| AutoJev-27B            | 80.5      | 80.5              | 0.82 / 0.77 / 0.94                            |
| Jev 1.13.0 (API)       | 80.1      | 78.8              | 1.73 / 0.69 / 2.37                            |
| Torchcast-decision-12B | 75.6      | 78.4              | 1.12 / 4.22 / 1.41                            |
| Winnow-12B             | 72.0      | 75.8              | 1.50 / 1.58 / 1.94                            |
| Cygnnet                | 75.3      | 75.6              | 1.09 / 1.19 / 1.50                            |
| Clef                   | 74.1      | 74.4              | 0.87 / 1.15 / 1.12                            |
| Plumb-4B               | 73.1      | 73.1              | 0.84 / 0.87 / 0.94                            |
| Open-Jev-27B           | 71.3      | 71.3              | 0.33 / 0.79 / 0.37                            |
| decider-4b-v2          | 70.8      | 70.8              | 0.97 / 0.97 / 0.77                            |
| SemIf                  | 58.9      | 62.7              | 1.30 / 1.30 / 1.41                            |
| Quyet-1.0-Medium       | 58.9      | 58.8              | 0.97 / 0.92 / 0.87                            |

Table 20. **Gemma 4 31B with and without fitted temperatures, by question type and subset** (OpenDecisions; raw / with temperatures; trained = mean of three seeds). Subsets: Capability of their items under the same scorer.

|                                   | Items | Zero-shot    | Trained     |
|-----------------------------------|-------|--------------|-------------|
| <i>Competence</i>                 |       |              |             |
| Choice                            | 1,178 | 84.5 / 84.5  | 86.4 / 86.4 |
| Yes/no                            | 586   | 77.0 / 77.0  | 78.4 / 78.4 |
| Score                             | 588   | 64.5 / 54.1  | 66.2 / 64.5 |
| <i>Calibration</i>                |       |              |             |
| Choice                            | 1,178 | 58.6 / 84.0  | 81.2 / 92.6 |
| Yes/no                            | 586   | 79.7 / 90.5  | 89.6 / 91.9 |
| Score                             | 588   | 72.9 / 87.6  | 89.9 / 92.4 |
| <i>Capability by label source</i> |       |              |             |
| Code                              | 599   | 71.8 / 73.9  | 81.4 / 81.3 |
| Authored                          | 1,443 | 72.3 / 78.9  | 80.5 / 83.1 |
| Real requests                     | 310   | 71.9 / 78.6  | 77.9 / 77.0 |
| <i>Capability by family</i>       |       |              |             |
| LLM routing                       | 694   | 76.3 / 81.9  | 82.7 / 83.6 |
| Multilingual                      | 422   | 70.6 / 77.4  | 80.0 / 82.5 |
| Fairness pairs                    | 236   | 62.2 / 63.9  | 65.0 / 65.6 |
| Use cases                         | 213   | 68.4 / 73.8  | 78.3 / 79.4 |
| Counting traps                    | 162   | 55.6 / 62.5  | 62.7 / 64.6 |
| Negation pairs                    | 160   | 100.0 / 94.3 | 99.9 / 99.6 |
| Option order                      | 120   | 77.2 / 77.7  | 83.7 / 82.2 |
| Perturbation pairs                | 110   | 48.9 / 52.0  | 60.8 / 63.2 |
| Composite rules                   | 106   | 73.2 / 83.4  | 83.5 / 83.0 |
| Calibration (generated)           | 94    | 72.9 / 69.6  | 84.7 / 82.1 |
| Calibration                       | 35    | 76.3 / 52.5  | 84.4 / 78.1 |
| All                               | 2,352 | 72.9 / 79.6  | 81.9 / 84.4 |

## H. Readout checks

**Probability outside the options.** The letter readout renormalizes the next-token probabilities of the  $K$  option letters (Eq. 1) and ignores all other tokens. For every zero-shot backbone the ignored mass is small: at most 0.7% on average on every set (Table 21). The most likely token is an option letter for almost every item. The exceptions are “Neither” for 9 items of OpenDecisions (Gemma 4 12B), and “NONE” or “None” for 7 questions of our validation set whose options do not fit the state (Gemma 4 31B).

Table 21. **Next-token probability outside the option letters**, zero-shot (full vocabulary softmax at the answer position, including Gemma’s soft-capping).

| Backbone    | OpenDecisions |                 |                         | JevBench-Open | Our validation set |
|-------------|---------------|-----------------|-------------------------|---------------|--------------------|
|             | Mean outside  | 99th percentile | Top token not an option | Mean outside  | Mean outside       |
| Gemma 4 31B | 0.01%         | 0.2%            | 0                       | 0.19%         | 0.24%              |
| Qwen3.6-27B | 0.37%         | 4.9%            | 0                       | 0.13%         | 0.67%              |
| Gemma 4 12B | 0.52%         | 6.1%            | 9                       | <0.01%        | 0.28%              |
| Qwen3.5-4B  | 0.09%         | 1.5%            | 0                       | 0.05%         | 0.05%              |

**Backbones without instruction tuning.** We also read out the pretrained versions of three backbones (google/gemma-4-31B, google/gemma-4-12B, Qwen/Qwen3.5-4B-Base; Table 22). Our prompt does not work for them: it uses the chat and thinking markers of the instruction-tuned models, which the pretrained models never saw. The pretrained Gemma 4 31B puts 99.9% of its next-token probability on tokens that continue them (“scratch”, “thought”, “thinking”, “system”) and scores 37.2. We therefore also use a plain completion prompt, as is standard for pretrained models: the same system text and JSON message as plain text, followed by “Answer:”, reading the tokens “ A”, “ B”, and so on. With this prompt, instruction tuning adds 14.9 points for Gemma 4 31B, 15.9 for Gemma 4 12B and 4.1 for Qwen3.5-4B. The plain prompt suits the instruction-tuned 12B and 4B models less well than our prompt (61.2 against 75.4, and 61.5 against 63.6), so the gap between a pretrained model and our zero-shot readout is larger still. Training the pretrained 31B model with our recipe (and our prompt) gives 70.3, below the instruction-tuned model without training (79.6).

Table 22. **Backbones with and without instruction tuning**, zero-shot letter readout with fitted temperatures (never below 1). Chat: our prompt (Appendix A). Plain: the same text without chat markers, followed by “Answer:”. Option mass: mean next-token probability on the option letters, on OpenDecisions. + recipe: trained with our recipe (400 steps, seed 1; the decision head reads out only the option letters).

| Backbone    | Version             | Prompt | Option mass | OpenDecisions |      |            | JevBench-Open |
|-------------|---------------------|--------|-------------|---------------|------|------------|---------------|
|             |                     |        |             | $I$           | $C$  | Capability | Accuracy (%)  |
| Gemma 4 31B | instruction-tuned   | chat   | 99.99%      | 71.9          | 87.4 | 79.6       | 91.8          |
|             | instruction-tuned   | plain  | 69.6%       | 67.8          | 90.0 | 78.9       | 89.6          |
|             | pretrained          | chat   | 0.13%       | 7.9           | 66.4 | 37.2       | 41.1          |
|             | pretrained          | plain  | 66.1%       | 42.1          | 85.9 | 64.0       | 76.6          |
|             | pretrained + recipe | chat   | —           | 58.4          | 82.1 | 70.3       | 81.0          |
| Gemma 4 12B | instruction-tuned   | chat   | 99.5%       | 64.0          | 86.8 | 75.4       | 86.1          |
|             | instruction-tuned   | plain  | 74.4%       | 43.4          | 79.1 | 61.2       | 72.3          |
|             | pretrained          | plain  | 66.1%       | 14.0          | 76.7 | 45.3       | 58.0          |
| Qwen3.5-4B  | instruction-tuned   | chat   | 99.9%       | 46.1          | 81.2 | 63.6       | 80.1          |
|             | instruction-tuned   | plain  | 30.1%       | 40.5          | 82.6 | 61.5       | 72.7          |
|             | pretrained          | plain  | 55.6%       | 38.6          | 76.3 | 57.5       | 71.0          |