When "A or B" Is Less Likely than A: Coherent Probabilities for Vision Decision Models

Seongrae Noh SeongEun Hong HyeongYeop Kang†

Korea University

{rhosunr99, seong_eun, siamiz_hkang}@korea.ac.kr

https://seongraenoh.github.io/core/

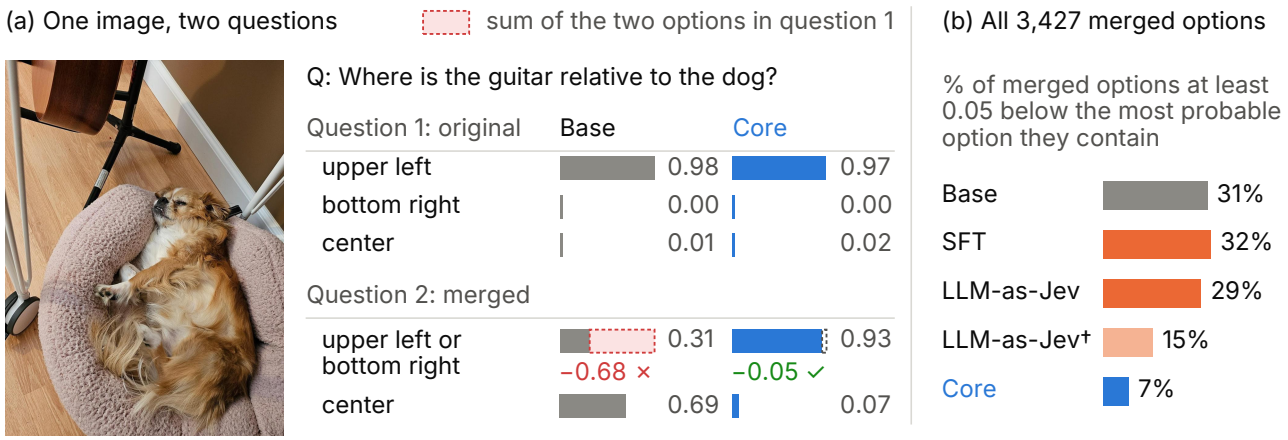


Figure 1. **A vision decision model assigns a merged option less probability than one of the options it contains.** (a) d3-lite on a RealWorldQA item (correct answer: upper left; option texts abbreviated; question 1 averaged over three option orders). Dashed boxes: the sum of the two options in question 1, the probability that coherence requires. (b) Share of the 3,427 merged options of the evaluation set that receive at least 0.05 less than the most probable option they contain (mean over seeds). LLM-as-Jev†: trained on our regrouped questions.

Abstract

Vision decision models answer a multiple-choice question about an image with one probability per option, and software uses them directly. We show that the probabilities depend on how the options are grouped: a merged option "A or B" receives on average 11.5 points less probability than the sum of A and B. This under-additivity appears at every model size from 0.85B to 26B and remains after fine-tuning on labels. We propose CORE*, which asks each training question with its options regrouped and trains the answers to all these questions toward one target distribution over the original options, summed into each question's groups. On 1,050 items from seven benchmarks,* CORE *reduces the total variation distance between the answer to a regrouped question and the original answer summed into the same groups from 14.8 to 5.8 points on the 0.85B model, against 9.4 for the best baseline trained on the same questions. At all five model sizes,* CORE *reduces this distance without lowering accuracy.*

1. Introduction

A decision model answers a question that has a fixed set of possible answers, such as a list of options, with one probability per answer instead of generated text. Jev [33] and the open d3 model family [34] are decision models, and vision decision models answer such questions about images [10, 20, 42]. Software uses the probabilities directly: it compares them with thresholds and combines them across questions [18]. Such uses assume that the probabilities are coherent: an event receives the same probability whether the question offers it as one option or as several.

Figure 1a shows that this assumption fails. Asked where the guitar is relative to the dog, with three options, the d3-

†Corresponding author.

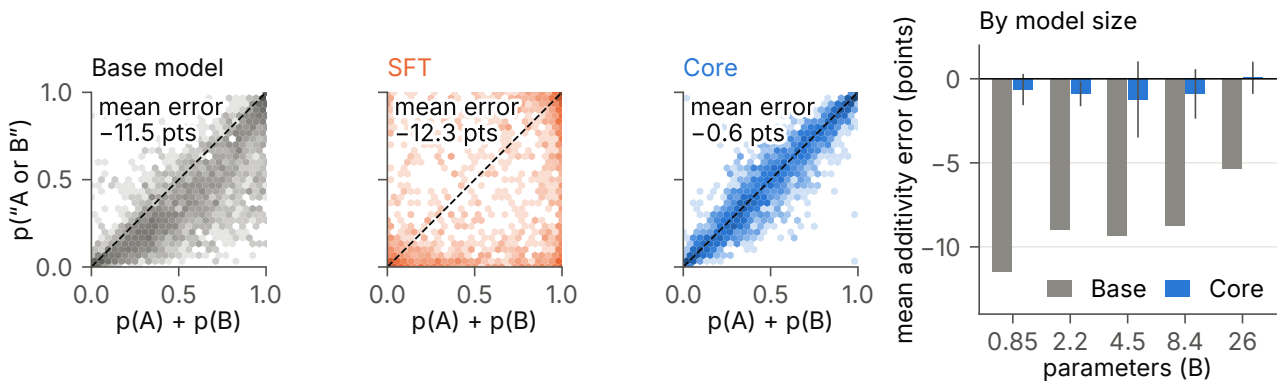


Figure 2. **Merged options receive less than the sum of their members, and fine-tuning on labels does not remove it.** The left three panels bin the 3,427 merged options of the d3-lite evaluation; darker hexagons hold more merged options (log scale). x: the sum of its members' probabilities on the original question; y: its probability on the merged question; dashed: perfect additivity. Right: mean additivity error in points for each model size (base: the model before fine-tuning; CORE: mean over seeds with 95% interval).

lite model assigns 0.98 to the correct option "upper left". When "upper left" and "bottom right" are merged into one option, the model assigns 0.31 to the merged option and picks "center". The merged option describes a larger event than "upper left", so its probability cannot be lower. Over the 3,427 merged options of our evaluation set, the base model assigns a merged option on average 11.5 points less than the sum of the options it contains, and in 31% of cases at least 5 points less than the most probable of them (Fig. 1b). We call this under-additivity. Human probability judgments show the same pattern, known as subadditivity: an event judged as a whole receives less probability than the sum of its parts judged separately [32].

Under-additivity does not depend on whether the model answers the question correctly. On the items whose original question the base model answers correctly, a merged option receives on average 11.0 points less than the options it contains, and 32% of merged options receive at least 5 points less than the most probable of them; on the items it answers wrongly, these figures are 12.2 points and 29%. Existing corrections address other errors. Temperature scaling [12] corrects the overall confidence level but leaves merged options as they are (partition violation, defined in Sec. 3: 14.8 → 14.6 points). Sorting the options or averaging over orders [10, 40] removes order sensitivity but cannot add probabilities across a merge. Consistency rewards for reinforcement learning (RL) [16, 41] require the chosen answers to agree across reordered questions, which constrains the chosen answer but not the probability of a merged option. Fine-tuning on labels raises accuracy but does not remove under-additivity: supervised fine-tuning (SFT) raises accuracy from 60.8 to 66.6, and a merged option still receives 12.3 points less than the options it contains (Fig. 2). Software could avoid the problem by asking the original question and adding the answers, but only if it knows which original options a merged option contains. When a user or another program writes the options, this information is not available, so the model itself must be coherent.

CORE (*coherence under regrouping*) trains the model to assign each event the same probability however the options are grouped. It asks each training question with its options reordered, merged into "X or Y", split into groups, with one option dropped, and with an option of another item added. All these questions share one target distribution over the original options: the frozen base model's answer mixed with the smoothed label. The target of each regrouped question is the same regrouping of this distribution: a merged option receives the sum of its members' target probabilities, dropping an option renormalizes the rest, and an added option receives almost none. A model that matches every target is therefore coherent on the training items. We optimize the targets with a sampled reward, the log-ratio of target to model probability, whose expectation is the negative KL divergence from the model's answer to the target. Optimizing the same targets exactly, without sampling, gives the same partition violation on d3-lite and d3-flash, so the targets, not the sampled reward, remove the violation.

Our contributions are summarized as follows:

- We show that vision decision models are under-additive: in the d3 models at every size, a merged option receives less probability than the sum of the options it contains, and fine-tuning on labels does not remove this (Sec. 3).
- We propose CORE, which trains a decision model toward one target distribution shared by all regroupings of a question (Sec. 4).
- On five d3 models from 0.85B to 26B, CORE lowers the partition violation without lowering accuracy, and the reduction remains for merge wordings not used in training and when CORE is trained on another dataset (Secs. 5.1, 5.3 and 5.5).

2. Related work

Decision models such as Jev [33], PixelJev [42], Releasing the Inner Jev [10] and LLM-as-Jev [20] return one probability per option. Their calibration is measured by the expected calibration error (ECE) [12, 23] and Brier score [11] on each question as it is asked, not on questions whose options are merged or split. Coherence requires that the probability of "A or B" equal the sum of the probabilities of A and B [2]; people violate this [32], and Li *et al*. [18] and Joy [17] report additivity violations of decision models on yes/no text questions and on hierarchical labels. We extend these diagnoses to merged options in questions about images and give a training method that reduces the violation. Order sensitivity, also called selection bias [25, 40], is corrected at inference by option sorting [10], one yes/no query per option [7] or an order-invariant architecture [19], and in training by RL rewards for answers that agree across option orders [16, 41]; other rewards based on group relative policy optimization (GRPO) [28] target calibrated confidence [1, 6, 38] or agreement across modalities [39]. These rewards score the sampled answer, whereas our reward compares its probability with a target probability. Our target is related to distillation targets [9, 14], and our regrouped questions to the augmentations of consistency regularization [29, 37]; unlike an augmentation, a regrouping changes the set of answers, so each regrouped question receives the regrouping of the target. Section A of the supplementary material discusses these works in full.

3. Incoherence under option regrouping

Problem setup. An item consists of images, a question and k mutually exclusive options $\Omega = \{\omega_1, \ldots, \omega_k\}$, one of which is correct. A decision model returns a distribution over the options as presented. We write $P(\omega)$ for the model's probability of ω on the original options, averaged over three random orders. A *regrouped question* f presents a list of blocks $B_1, \ldots, B_m$: disjoint sets of original options, each shown as one option whose text joins its members with "or". We use five kinds of regrouping, each drawn without looking at the correct answer: a new *order* of the original options; a *pair merge* of two random options; a random *split* of all options into 2 to $k-1$ groups; a *drop* of one option; and an *add* of one option borrowed from another item of the same benchmark. The model's answer to f is a distribution p_f over the presented blocks.

Coherence constraints. Because the options are mutually exclusive, a block is the union of its members, and its probability should be their sum:

$$p_f(B) = \textstyle\sum_{\omega \in B} P(\omega). \qquad (1)$$

Equation (1) covers orders (blocks of size one) and merges. When an option is dropped, the remaining options should keep their ratios, $p_f(\omega) = P(\omega) / \sum_{\omega' \in f} P(\omega')$. Adding an option changes the set of possible answers, so the probabilities need not stay the same in general [42]. We therefore add only options taken from other items, treat them as wrong, and report the add case separately. We measure the violation of Eq. (1) by the *partition violation*

$$V(f) = \tfrac{1}{2} \textstyle\sum_{B \in f} \left| p_f(B) - \sum_{\omega \in B} P(\omega) \right|,$$

the total variation (TV) distance between the answer to a regrouped question and the original answer summed into the same blocks, averaged over the pair merges and splits of all items. Order TV is the mean TV distance between the answers to the three orders. IIA TV (independence of irrelevant alternatives) is the TV distance between the answer to a drop or add question and the original answer, both restricted to the original options they share and renormalized. We also report the *additivity error* $p_f(B) - \sum_{\omega \in B} P(\omega)$ of every merged option (a block with two or more members), in points ($\times 100$); a negative error means under-additivity. All metrics are reported $\times 100$.

Evaluation data. We evaluate on 1,050 items, 150 from each of seven benchmarks of the Jev Decision Index vision suite [3]: CV-Bench [30], BLINK [8], RealWorldQA [36], CharXiv [35], InfographicVQA [22], Mind2Web [5] and R-Bench [13], with 3 to 6 options. Each item yields 8 questions (3 orders, 2 pair merges, 1 split, 1 drop, 1 add; Mind2Web items have no add), 8,250 questions with 3,427 merged options in total. Training items (2,310) come from the same benchmarks and share no item with the evaluation set (details in Sec. B of the supplementary material). For each model, the *hard* items are the 25% on which the base model's mean violation over the five kinds of regrouping (order TV, partition violation of the merges and of the split, IIA TV of the drop and the add) is highest; every method is evaluated on this same set. Because the set is selected by the base model's own violations, part of any trained model's improvement over the base model on it is regression to the mean; comparisons between trained methods are not affected.

Empirical findings. Figure 2 shows the answer of d3-lite to every merged question against the sum of the answers to the original question. The base model assigns a merged option on average 11.5 points less than the sum of its members, and 31% of merged options receive at least 5 points less than their most probable member, which no coherent assignment allows. The mean additivity error is negative at every size of the d3 family: -9.0 points at 2.2B, -9.3 at 4.5B, -8.7 at 8.4B and -5.4 at 26B; the 26B model is less affected but not coherent. Generative VLMs scored by the probability

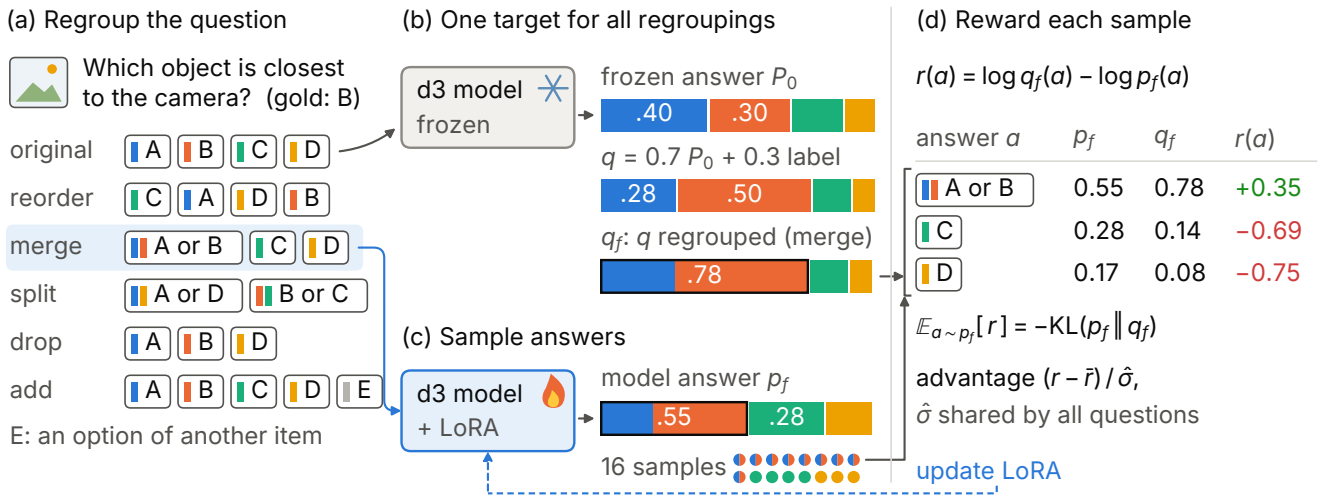


Figure 3. **CORE in one step.** (a) A training item is asked as its original question and as regrouped questions. (b) The frozen model's answer to the original question, mixed 0.7 to 0.3 with the smoothed label, gives one target q over the original options; each regrouped question gets q regrouped the same way. (c) The trained model answers the regrouped question, and 16 answers are sampled. (d) The reward of each sample is the log-ratio of its target and model probability; advantages are divided by one reward scale $\hat{\sigma}$ shared across questions, and the LoRA weights are updated. Shown for the question with a merged option; numbers are illustrative.

of the option letter (Qwen3-VL-Instruct, 2B to 8B) are less coherent still: their partition violation is 21.7 to 23.8, much of it from order sensitivity (order TV 22.8 to 24.5), and 22.9 to 31.2% of their merged options receive at least 5 points less than their most probable member (Tab. S6 in the supplementary material). We define the strength of an option as its probability divided by the total probability of the unmerged options. Of four predictions of a merged option's strength from its members' strengths, their sum, which coherence requires, is the worst at every size, and their geometric mean is the best on d3-lite (mean absolute log error 0.54 against 1.13; Tab. S2). A merged option is predicted better by an average of its members than by their sum.

Fine-tuning on labels does not remove under-additivity. SFT on the original options raises accuracy by 5.8 points, but its mean additivity error is −12.3 points and its partition violation rises from 14.8 to 19.3: the model becomes more confident on each question separately, and a merged option that the model does not pick receives near-zero probability (Fig. 2, SFT panel). LLM-as-Jev, which adds a KL anchor to the frozen model, keeps the mean additivity error at −11.3 points. Training-free corrections do not remove it either: temperature scaling changes the partition violation from 14.8 to 14.6, and asking one yes/no question per option (OBS) lowers it to 11.6 at k times the inference cost (Sec. 5.2). We therefore train on regrouped questions directly.

4. CORE: Coherence under regrouping

CORE fine-tunes a decision model so that its answers to regrouped questions satisfy Eq. (1). It has three parts: a target for every regrouped question (Sec. 4.1), a reward that moves each answer toward its target (Sec. 4.2), and a reward scale shared across questions (Sec. 4.3). The ablations in Sec. 5.4 show that the target is the part that reduces the violation. Figure 3 shows one step.

4.1. Regrouping-consistent targets

Each training item is asked in two fixed random orders of its original options. Let P_0 be the frozen base model's distribution over the original options, averaged over these two orders (unlike P in Sec. 3, which is the evaluated model's own answer averaged over three evaluation orders), and let $h_\epsilon = (1-\epsilon)\,h + \epsilon/k$ be the one-hot label h smoothed with $\epsilon = 0.05$. For a distribution X over the original options and a block B we write $X(B) = \sum_{\omega\in B} X(\omega)$. For a regrouped question f with blocks B, the target is

$$q_f(B) = w\,\frac{h_\epsilon(B)}{\sum_{B'\in f} h_\epsilon(B')} + (1-w)\,\frac{P_0(B)}{\sum_{B'\in f} P_0(B')}, \quad (2)$$

with label weight $w = 0.3$. A borrowed option in an add question has no member in Ω; it receives mass ϵ/k in the label term and 10^{-3} in the frozen-model term before normalization. For every question that keeps all original options, the denominators are one and $q_f(B) = \sum_{\omega\in B} q(\omega)$ with $q = w\,h_\epsilon + (1-w)P_0$: all targets of an item are regroupings of one distribution q, so a model that matches every target satisfies Eq. (1) on the training items. The frozen-model term keeps the target close to the frozen model's answer to the original question; it needs no labels and carries the model's own uncertainty. The label term moves the target toward the correct answer, so that CORE can become more accurate than the base model (Sec. 5.4).

4.2. Log-ratio reward

For each question f the model samples $K = 16$ answers $a_1, \ldots, a_K \sim p_f$ and receives

$$r_i = \log q_f(a_i) - \log p_f(a_i), \qquad \mathbb{E}_{a \sim p_f}[r] = -\mathrm{KL}(p_f \,\|\, q_f)\,. \tag{3}$$

An answer that the model under-weights relative to its target, such as a merged option in Fig. 3d, receives a positive reward, and an answer that it over-weights receives a negative reward. The loss of question f is the policy-gradient loss $\mathcal{L}_f = -\frac{1}{K}\sum_i A_i \log p_f(a_i)$ with advantages $A_i = (r_i - \bar{r})/\hat{\sigma}$, where $\bar{r}$ is the mean reward of the K samples. Each step uses four training items. Each item is asked as four questions: its two training orders and two regroupings drawn uniformly from its merges, split, drop and add. The step loss is the mean over the 16 questions. Because the reward depends on the full distribution through $\log p_f$, the expected update reduces the KL divergence to the target rather than raising the probability of one preferred answer. A reward of $\log q_f(a)$ alone is maximized by putting all mass on the block with the largest target, not by $p_f = q_f$. In d3-lite pilot runs with the reward $\log q_f(a)$ alone, merged options without the correct answer received 3.7 to 5.2 times the sum of their members' probabilities (Tab. S3).

4.3. Advantage normalization and KL regularization

GRPO divides each question's rewards by their own standard deviation [28], which gives a question that already matches its target the same advantage scale as one far from it. We divide by one running estimate over all questions, $\hat{\sigma} \leftarrow 0.99\,\hat{\sigma} + 0.01\,\mathrm{std}(r_{1:K})$ after a warm-up of 50 questions, so that questions far from their target receive larger advantages; on d3-lite the difference from GRPO's scale is within seed variation (Sec. 5.4). If all K samples of a question fall in one block, the advantages are zero and the question gives no update. We add no KL term $\mathrm{KL}(P_0 \| \bar{p})$ toward the frozen model [16, 20], where $\bar{p}$ is the trained model's answer on the original options: the target already contains P_0 with weight 0.7.

Implementation details. A d3 model reads the final hidden state of its prompt encoder and maps it to one score per presented option with a readout matrix; a softmax over these scores gives p_f. We train LoRA adapters [15] of rank 16 ($\alpha = 32$) on all linear layers of the language model; the vision encoder and the readout matrix are frozen. We use AdamW with learning rate $3 \cdot 10^{-5}$, no schedule, gradient clipping at 1.0 and bf16, for 600 steps of 4 items (300 steps for the 26B model; all hyperparameters in Sec. C of the supplementary material). A d3-lite run takes 42 minutes on one RTX 4090, a d3-flash run about 70 minutes on one H100.

5. Experiments

Models. We fine-tune the five d3 decision models [34]: d3-lite (0.85B parameters), d3-nano (2.2B), d3-mini (4.5B), d3-flash (8.4B) and d3 (26B). Baselines are compared on d3-lite and d3-flash.

Baselines. Every baseline is fine-tuned on the same training items as CORE: SFT with cross-entropy, as in PixelJev and Releasing the Inner Jev [10, 42]; LLM-as-Jev [20], cross-entropy plus a KL anchor to the frozen model on the option distribution (learning rate 10^{-4} as published); ACRE [16] and PA-GRPO [41], RL with rewards for correct and order-consistent answers. On d3-lite we train each baseline in two conditions: on the original question in two orders, as published, and on the same regrouped questions as CORE (marked †); on d3-flash, all four as published and SFT and LLM-as-Jev also on the regrouped questions. All methods use the same number of steps and LoRA rank. The learning rates of ACRE and PA-GRPO are listed in Sec. D of the supplementary material.

Evaluation protocol. Baselines use 3 training seeds; CORE uses 5 on d3-lite and 3 on d3-flash. We report means and standard deviations over seeds; 95% intervals over seeds are in Sec. H of the supplementary material, together with paired item-bootstrap intervals for every difference from CORE. Accuracy (Acc.) is the fraction of questions, original and regrouped, whose most probable block contains the correct option. Acc., ECE (15 bins) and the Brier score use all questions in which some block contains the correct option.

Training-free baselines. Methods that do not change the model's weights learn nothing from the training items, so we compare them separately (Sec. 5.2): temperature scaling [12], the calibration step of PixelJev and Releasing the Inner Jev [10, 42], with one temperature fitted on 400 training items; options sorted by text, the order fix of Releasing the Inner Jev; averaging the answers to the given and the reversed order, an option of the d3 runtime [34]; and OBS [7], one yes/no question per option. Recovering the members of a merged option from its text and summing their probabilities on the original question is compared with CORE in Sec. I of the supplementary material.

5.1. Main results

Table 1 gives the metrics of every trained method on d3-lite, and Fig. S1 plots accuracy against partition violation at both sizes. On d3-lite, CORE lowers the partition violation from 14.8 to 5.8 and on the hard items from 25.7 to 8.5, and it lowers ECE from 5.6 to 1.7. Trained on the original options, no baseline lowers the partition violation by more than 0.4 points, and SFT raises it to 19.3: cross-entropy raises the top

Table 1. **Trained methods on d3-lite.** Acc.: accuracy over all questions, original and regrouped; ECE and Brier score over all questions; partition violation on all items and on the hardest 25%; order TV; IIA TV when an option is added or dropped (all $\times 100$). Mean $\pm$ standard deviation over 3 seeds per baseline and 5 for CORE. Bold: best; underline: second best; the base model is not ranked.

	Acc. ↑	ECE ↓	Brier ↓	Partition ↓		Order ↓	IIA ↓	
Method		all	all	all	hard	TV	add	drop
Base model	60.8	5.6	48.4	14.8	25.7	5.5	5.5	6.4
SFT [10, 42]	66.6 ±1.2	21.3 ±1.9	52.5 ±0.3	19.3 ±1.0	27.8 ±1.5	9.5 ±0.3	10.4 ±0.3	12.3 ±0.5
LLM-as-Jev [20]	66.8 ±0.5	1.9 ±0.6	42.9 ±0.8	14.4 ±0.8	21.2 ±1.4	6.3 ±0.3	7.1 ±0.4	7.4 ±0.2
ACRE [16]	66.4 ±0.6	7.4 ±1.8	44.6 ±1.4	15.9 ±1.7	23.7 ±1.7	10.0 ±2.8	8.5 ±1.6	10.3 ±2.6
PA-GRPO [41]	63.8 ±0.3	16.8 ±0.8	51.9 ±1.2	16.0 ±1.3	24.6 ±1.4	10.6 ±2.6	9.9 ±2.2	11.4 ±1.4
CORE (ours)	**67.1** ±0.6	**1.7** ±0.1	**42.4** ±0.5	**5.8** ±0.1	**8.5** ±0.2	**4.1** ±0.2	**4.5** ±0.2	**5.2** ±0.2

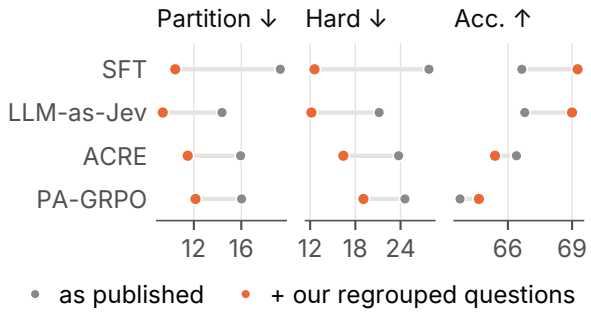


Figure 4. **Baselines trained as published and on our regrouped questions, d3-lite.** Gray: trained on the original question, as published; orange: also trained on our regrouped questions (†). Partition violation on all and on hard items, and Acc. ($\times 100$; means over 3 seeds).

probability of each question separately, and nothing in it constrains merged options. Training on our regrouped questions lowers the partition violation of every baseline on d3-lite (Fig. 4), so the regrouped questions reduce the violation even without our target. The best of these baselines, LLM-as-Jev†, reaches 9.4 (hard items: 12.2), which is 3.5 (3.6) points above CORE. CORE's ECE is not significantly lower than that of LLM-as-Jev, and LLM-as-Jev† has a lower Brier score (40.7 against 42.4; paired intervals in Tab. S8). In accuracy, CORE is 6.3 points above the base model and within the paired 95% intervals of SFT, LLM-as-Jev and ACRE trained on the original options; the cross-entropy baselines trained on the regrouped questions are more accurate, LLM-as-Jev† by 1.8 and SFT† by 2.1 points. No baseline, with or without the regrouped questions, has both a lower partition violation and a higher accuracy than CORE (Tab. 1 and Fig. 4).

Residual violation and downstream decisions. Order sensitivity accounts for part of the remaining violation: P averages three orders while each regrouped question is asked in one, and a model that satisfies Eq. (1) exactly but is as order-sensitive as CORE would have a partition violation of 2.4 instead of 5.8. At a threshold of 0.5, the decision from a merged option differs from the one implied by the original question for 18.6%, 8.4% and 7.4% of merged options with the base model, LLM-as-Jev† and CORE. CORE's partition violation is lower than that of the base model and LLM-as-Jev† on each of the seven benchmarks (Tab. S5).

Results on d3-flash. CORE again has the lowest partition violation (Tab. S9): 4.8 against 15.0 for the base model (hard items: 9.2 against 35.7), with an accuracy gain of 4.6 points. The closest baseline, SFT†, reaches 7.3 (hard items: 14.9) at an ECE of 12.0 against 2.3; SFT and LLM-as-Jev, with and without the regrouped questions, have a higher mean accuracy than CORE (78.0 to 80.7 against 77.2), and ACRE and PA-GRPO a lower one. Unlike on d3-lite, the regrouped questions do not lower the partition violation of LLM-as-Jev (12.1 with them, 11.5 without). Both versions have an ECE not significantly different from CORE's and a lower Brier score (28.1 and 29.3 against 30.2).

5.2. Comparison with training-free corrections

We test whether the violation can be corrected at inference (Tab. 2): by correcting confidence or option order, or by asking about each option separately. Temperature scaling lowers ECE (5.6 to 1.9 on d3-lite), sorting removes order sensitivity by construction, and averaging two orders reduces them; none of the three changes the partition violation by more than 0.9 points on d3-lite and 1.1 on d3-flash. OBS lowers the partition violation to 11.6 on d3-lite and 11.6 on d3-flash at k times the inference cost, and raises ECE on d3-flash (3.5 to 11.7; 5.6 to 6.4 on d3-lite). Correcting confidence or option order does not reduce the violation; OBS reduces it at k times the inference cost, and CORE at no extra inference cost.

5.3. Effect of model size

Figure 5 compares CORE with the base model at all five sizes. The base models' partition violation lies between 10.2 and 16.4 and does not fall with size up to 8.4B; on hard

Table 2. **Training-free corrections.** Temperature scaling fits one temperature on 400 training items; order averaging averages the answers to two option orders; OBS asks one yes/no question per option. CORE (trained; mean ± standard deviation over seeds) is shown for reference. Columns as in Tab. 1; bold: best; underline: second best.

	Acc. ↑	ECE ↓	Brier ↓	Partition ↓		Order ↓	IIA ↓	
Method		all	all	all	hard	TV	add	drop
d3-lite (0.85B)								
Base model	60.8	5.6	48.4	14.8	25.7	5.5	5.5	6.4
Temperature scaling [12]	60.8	1.9	47.9	14.6	23.3	4.7	4.9	5.6
Sorted options [10]	61.1	5.9	48.1	13.9	22.9	**0.0**	4.1	5.5
Order averaging (2×) [34]	61.3	4.7	48.1	14.2	24.2	2.9	**4.0**	**5.1**
OBS (k×) [7]	61.3	6.4	49.7	11.6	15.0	5.3	5.6	5.5
CORE (trained)	**67.1** ±0.6	**1.7** ±0.1	**42.4** ±0.5	**5.8** ±0.1	**8.5** ±0.2	4.1 ±0.2	4.5 ±0.2	5.2 ±0.2
d3-flash (8.4B)								
Base model	72.6	3.5	35.3	15.0	35.7	5.9	6.7	12.2
Temperature scaling [12]	72.6	2.7	35.1	14.3	31.1	5.4	6.2	10.6
Sorted options [10]	74.2	4.3	34.2	13.9	31.6	**0.0**	4.7	11.7
Order averaging (2×) [34]	73.9	**2.0**	34.7	14.5	33.4	3.3	5.1	10.9
OBS (k×) [7]	74.7	11.7	37.0	11.6	15.1	6.5	6.5	7.0
CORE (trained)	**77.2** ±0.2	2.3 ±0.3	**30.2** ±0.2	**4.8** ±0.2	**9.2** ±0.4	3.8 ±0.1	**4.0** ±0.1	**5.0** ±0.1

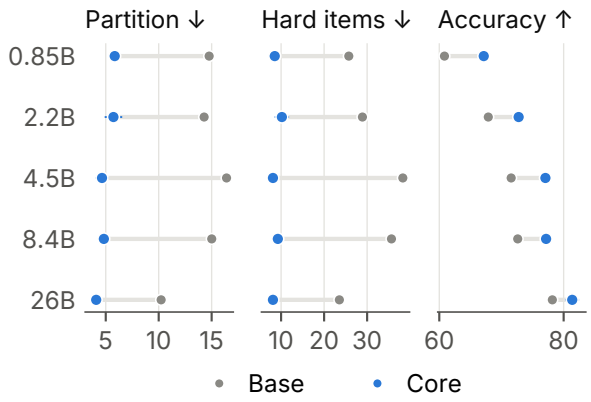


Figure 5. **Base model and CORE at five model sizes.** Partition violation on all and on hard items, and Acc. (×100). Gray: base model; blue: CORE (mean over seeds with 95% interval).

items it is highest at 4.5B and 8.4B (38.3 and 35.7). CORE lowers it to 4.1–5.8 (hard items: 8.1–10.2) at every size and raises accuracy by 3.2 to 6.3 points. ECE falls at 0.85B and 2.2B, does not change significantly at 4.5B and 8.4B, and rises from 1.7 to 3.2 at 26B, where the base model is the best calibrated of the five (paired intervals in Sec. H of the supplementary material).

5.4. Ablation study

Table 3 changes one part of CORE at a time on d3-lite. Adding the KL anchor to the frozen model gives the same partition violation (5.8) and an accuracy of 66.9 against 67.1, a difference within seed variation. Dividing each question's rewards by their own standard deviation, as GRPO does, gives an accuracy of 67.1 and a partition violation of 6.1 against 67.1 and 5.8; these paired differences are within seed variation (Tab. S13). The label weight w trades accuracy against calibration and consistency. Without the label ($w = 0$) the target is the frozen model alone: the partition violation is 5.5, slightly below CORE's, but accuracy is 64.3 against 67.1. With $w = 0.5$ accuracy is 68.6 (the 95% interval of the paired difference includes zero), and ECE and partition violation rise to 2.5 and 6.4. With the label alone ($w = 1$) the target no longer carries the model's uncertainty: ECE rises to 13.3 and partition violation to 11.8, without a further gain in accuracy (68.2). Raising w from 0 to 0.6 during training gives an accuracy of 68.2 at an ECE of 3.2.

Table 3. **Ablations on d3-lite.** Each row changes one part of CORE (×100; mean ± standard deviation over 3 seeds; the CORE row is taken from Tab. 1). The last two rows optimize the same target without sampling.

	Acc.↑	ECE↓	Part.↓	Hard↓	Order↓
CORE	67.1 ±0.6	1.7 ±0.1	5.8 ±0.1	8.5 ±0.2	4.1 ±0.2
+ KL anchor	66.9 ±0.2	1.4 ±0.2	5.8 ±0.1	8.7 ±0.3	4.1 ±0.1
per-question std	67.1 ±0.4	2.0 ±0.1	6.1 ±0.3	9.0 ±0.4	4.1 ±0.0
$w = 0$ (no label)	64.3 ±0.3	2.5 ±0.2	5.5 ±0.1	8.3 ±0.4	3.8 ±0.1
$w = 0.5$	68.6 ±0.2	2.5 ±0.2	6.4 ±0.2	8.9 ±0.3	4.4 ±0.1
$w = 1$ (label only)	68.2 ±0.7	13.3 ±1.3	11.8 ±0.6	13.1 ±0.8	8.0 ±0.1
w from 0 to 0.6	68.2 ±0.3	3.2 ±0.7	6.5 ±0.1	8.9 ±0.5	4.3 ±0.1
exact $\mathrm{KL}(p_f \| q_f)$	67.3 ±0.3	1.5 ±0.4	5.8 ±0.1	8.5 ±0.2	4.0 ±0.1
cross-entropy to q_f	67.8 ±0.2	2.0 ±0.8	5.7 ±0.1	8.5 ±0.2	3.9 ±0.1

Sampled versus exact objectives. A decision model gives its full answer distribution in one forward pass, so the expec-

Table 4. **Partition violation under unseen merge wordings, d3-lite** ($\times 100$; lower is better). The merged options of the evaluation set are worded in ways not used in training (training uses "X or Y"); CORE: mean $\pm$ standard deviation over 2 seeds.

	X or Y	one of	either	any of	X / Y
Base model	14.8	14.1	15.1	15.1	17.9
CORE	5.8 $\pm$0.1	6.6 $\pm$0.1	6.7 $\pm$0.3	7.0 $\pm$0.3	11.7 $\pm$0.2

Table 5. **Training on A-OKVQA, evaluated on the Decision Index items, d3-lite** ($\times 100$). Mean $\pm$ standard deviation over 3 seeds; the last row is CORE from Tab. 1.

Method	Training items	Acc.↑	ECE↓	Part.↓	Hard↓
Base model	–	60.8	5.6	14.8	25.7
SFT	A-OKVQA	61.9 $\pm$0.1	17.5 $\pm$3.8	12.0 $\pm$0.3	20.7 $\pm$0.5
CORE	A-OKVQA	62.2 $\pm$0.4	2.4 $\pm$0.6	8.3 $\pm$0.0	14.8 $\pm$0.2
CORE	Decision Index	67.1 $\pm$0.6	1.7 $\pm$0.1	5.8 $\pm$0.1	8.5 $\pm$0.2

tation in Eq. (3) can also be computed exactly. The last two rows of Tab. 3 optimize the same target without sampling: by the exact $\mathrm{KL}(p_f \| q_f)$ over the blocks, and by the cross-entropy $-\sum_B q_f(B) \log p_f(B)$. They reach accuracies of 67.3 and 67.8 and partition violations of 5.8 and 5.7, against 67.1 and 5.8 for CORE; on d3-flash the exact KL reaches 77.4 and 4.6 against 77.2 and 4.8. In ECE and partition violation, every paired difference to CORE is within its 95% interval over seeds, and in accuracy all but one: the exact KL on d3-flash is slightly more accurate (Tab. S14). The target of Eq. (2), not the sampled estimator, reduces the violation. The sampled reward needs only the probabilities of the answers it draws, so it also applies when the answer set is too large to sum over (tested in Sec. J of the supplementary material); with the 3 to 6 options of our items, the exact objectives work as well.

5.5. Generalization and control analyses

Control for confidence. A model can satisfy Eq. (1) by answering uniformly, so a lower partition violation could come from lower confidence [18]. CORE's mean top probability on the original options (63.8) is within 0.1 points of the base model's (63.7). In each of four bins of each model's own top probability, CORE's partition violation (1.9 to 7.8) is lower than the base model's (9.3 to 17.4) and SFT's (Tab. S15).

Unseen merge wordings. Training uses "X or Y" only. Table 4 rewords the merged options of the evaluation set in four other ways. On "one of", "either" and "any of", CORE's violation is 6.6, 6.7 and 7.0, close to the trained wording's 5.8 and less than half of the base model's (14.1 to 15.1). On "X / Y", which has no connective, it falls from 17.9 to 11.7 (by 35%, against 53–61% on the other wordings).

Cross-dataset training. Trained on A-OKVQA [27] items (Tab. 5), which have COCO images and everyday questions, and evaluated on the Decision Index items, CORE lowers the partition violation from 14.8 to 8.3 (hard items: 25.7 to 14.8) with an ECE of 2.4; SFT on the same items reaches 12.0 with an ECE of 17.5. Accuracy stays near the base model for both (62.2 and 61.9 against 60.8). The reduction in partition violation transfers from A-OKVQA training items to the Decision Index items, whereas the accuracy gain of Sec. 5.1 requires training items from the evaluation benchmarks.

Dependence on the image. If CORE became coherent by ignoring the image, removing the image would leave its accuracy intact. We test this with the seed-0 model of CORE, whose accuracy with the original image is 67.8. Replacing the image with a blank one lowers this accuracy by 29.5 points (to 38.3) and the base model's by 27.3; with the image of another item, CORE reaches 37.9 and the base model 33.3. CORE's accuracy depends on the image as much as the base model's. Its partition violation rises from 5.8 to 7.6 and 6.8, still below the base model's 12.6 and 14.7.

Generative VLMs. Section K of the supplementary material applies CORE to generative VLMs scored by the probability of the option letter.

6. Conclusion

Vision decision models assign a merged option less probability than the sum of its members, often less than its most probable member, at every size we tested, and fine-tuning on labels does not remove this. CORE trains the answers to all regrouped questions of an item toward regroupings of one target distribution, with a reward whose expectation is the negative KL divergence to that target. The target, not the sampled reward, reduces the violation; CORE also raises accuracy at every size and lowers ECE at 0.85B and 2.2B. Regrouped questions are generated automatically and need no labels to evaluate, so we suggest reporting partition violation next to accuracy and ECE for decision models.

Limitations. We test only merged options that list their members. Merged options that name a broader event without listing its members ("left half") are future work.

LLM Usage Statement

We used Claude (Anthropic) through Claude Code to assist the authors with the experiments and the writing of this paper. The authors are fully responsible for the content.

References

[1] Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond

binary rewards: Training lms to reason about their uncertainty. In *International conference on learning representations*, pages 59456–59493, 2026. 3, 11

[2] Bruno De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, pages 1–68, 1937. 3, 11

[3] Decision Index maintainers. Jev Decision Index. Leaderboard: https://huggingface.co/spaces/multimodalart/jev-decision-index. Reproduction kit: https://github.com/apolinario/decision-index, 2026. Accessed November 15, 2026. 3, 11

[4] Julien Delaunay. Can the confidence and decisions of system 1 models be trusted? an explainability and perturbation audit of jev-style decision models. *arXiv preprint arXiv:2611.12521*, 2026. 11

[5] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023. 3

[6] Yuetian Du, Yucheng Wang, Rongyu Zhang, Zhijie Xu, Boyu Yang, Ming Kong, Jie Liu, and Qiang Zhu. Linking perception, confidence and accuracy in mllms. *arXiv preprint arXiv:2603.12149*, 2026. 3, 11

[7] Yanrui Du, Danyang Zhao, Sendong Zhao, Wenhao Li, Yibo Gao, Haochun Wang, Zhuoer Xu, Shiwen Cui, and Bing Qin. How reliable are system-one decisions in medicine? a comparative study of jev, jev-like models, and llms. *arXiv preprint arXiv:2610.18529*, 2026. 3, 5, 7, 11, 13

[8] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024. 3

[9] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018. 3, 12

[10] Jonathan Geuter, Andrew Cheng, and David Alvarez-Melis. Releasing the inner jev: A simple recipe for open decision models. *arXiv preprint arXiv:2610.17205*, 2026. 1, 2, 3, 5, 6, 7, 11, 13

[11] W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. 3, 11

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 2, 3, 5, 7, 11, 13

[13] Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, et al. R-bench: Graduate-level multi-disciplinary benchmarks for llm & mllm complex reasoning evaluation. *arXiv preprint arXiv:2505.02018*, 2025. 3

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 12

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5, 11

[16] Minbin Huang, Runhui Huang, Chuanyang Zheng, Jingyao Li, Guoxuan Chen, Han Shi, and Hong Cheng. Answer-consistent chain-of-thought reinforcement learning for multi-modal large langauge models. *arXiv preprint arXiv:2510.10104*, 2025. 2, 3, 5, 6, 11

[17] Saman Sarker Joy. Do system one decisions add up? a study of probabilistic coherence. *arXiv preprint arXiv:2609.33971*, 2026. 3, 11

[18] Keyi Li, Yihao He, and Quanyi Li. Beyond calibration: Do a typed-decision model's probabilities obey the probability axioms? *arXiv preprint arXiv:2609.33209*, 2026. 1, 3, 8, 11

[19] Qi Li and Xinchao Wang. Jeva: System one models from first principles. *arXiv preprint arXiv:2610.18773*, 2026. 3, 11

[20] Yinheng Li and Justin Wagle. Llm2jev: Llms are already jev-style decision models–when and how to fine-tune them. *arXiv preprint arXiv:2610.02076*, 2026. 1, 3, 5, 6, 11

[21] Yesheng Liu, Hao Li, Haiyu Xu, Baoqi Pei, Jiahao Wang, Mingxuan Zhao, Jing-Shu Zheng, Zheqi He, JG Yao, Xi Yang, et al. Beyond multiple choice: Verifiable openqa for robust vision-language rft. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18859–18869, 2026. 11

[22] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2582–2591. IEEE, 2022. 3

[23] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, 2015. 3, 11

[24] Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019. 25

[25] Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, 2024. 3, 11

[26] Muhammad Atif Saeed, Azka Atiq, Adarsh Kumar, Akhtar Jamil, Uzair Iqbal, and Roland Aydin. Shift-jev: Certified abstention and gain-gated continual adaptation for llm-based decision models under distribution shift. *arXiv preprint arXiv:2610.34329*, 2026. 11

[27] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022. 8

[28] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3, 5, 11

[29] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 3, 12

[30] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024. 3

[31] Amos Tversky and Daniel Kahneman. Rational choice and the framing of decisions. In *Multiple criteria decision making and risk analysis using microcomputers*, pages 81–126. Springer, 1989. 11

[32] Amos Tversky and Derek J Koehler. Support theory: A nonextensional representation of subjective probability. *Psychological review*, 101(4):547, 1994. 2, 3, 11

[33] TypeSafe AI. Introducing System One models & Jev. TypeSafe AI blog, https://typesafe.ai/blog/introducing-system-one-models-and-jev, 2026. Published September 15, 2026. Accessed November 15, 2026. 1, 3, 11

[34] vLLM Semantic Router Team. d3: A multimodal foundation decision model. Hugging Face model card, https://huggingface.co/vllm-sr/d3, 2026. Model family vllm-sr/d3-lite, d3-nano, d3-mini, d3-flash and d3. Accessed November 15, 2026. 1, 5, 7, 13

[35] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024. 3

[36] xAI. Grok-1.5 Vision Preview. https://x.ai/news/grok-1.5v, 2024. Release of the RealWorldQA benchmark, April 12, 2024. 3

[37] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020. 3, 12

[38] Hee Suk Yoon, Eunseop Yoon, Ji Woo Hong, SooHwan Eom, Gwanhyeong Koo, Mark Hasegawa-Johnson, Qi Dai, Chong Luo, and Chang D Yoo. Pdcr: Perception-decomposed confidence reward for vision-language reasoning. *arXiv preprint arXiv:2605.13467*, 2026. 3, 11

[39] Zirui Zhang, Haoyu Dong, Kexin Pei, and Chengzhi Mao. R-c2: Cycle-consistent reinforcement learning improves multimodal reasoning. *arXiv preprint arXiv:2603.25720*, 2026. 3, 11

[40] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, pages 19426–19454, 2024. 2, 3, 11

[41] Jinquan Zheng, Jia Yuan, Jiacheng Yao, Chenyang Gu, Pujun Zheng, and Guoxiu He. Mitigating selection bias in large language models via permutation-aware grpo. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35125–35143, 2026. 2, 3, 5, 6, 11

[42] Xunlan Zhou, Xianliang Yang, and Li Zhao. From text decisions to pixels: An study of jev-style visual choice model. *arXiv preprint arXiv:2609.29283*, 2026. 1, 3, 5, 6, 11, 13

When "A or B" Is Less Likely than A: Coherent Probabilities for Vision Decision Models

Supplementary Material

Seongrae Noh SeongEun Hong HyeongYeop Kang

Contents

A. Extended related work 11

B. Data and regrouping details 12

C. Implementation details 12

D. Baseline details 13

E. Predicting a merged option from its members 13

F. Pilot study: target-probability reward 14

G. Additional analyses 14

H. Full results with confidence intervals 16

I. Comparison with parsing merged options 19

J. Sampled reward on large answer sets 20

K. CORE on generative VLMs 21

L. Qualitative examples and failure cases 22

M. Robustness of the metrics 25

N. Computational cost 25

A. Extended related work

Decision models and calibration. Decision models such as Jev [33] return a probability for each option of a question with a fixed set of answers instead of generating text, and public leaderboards rank them by accuracy and calibration [3]. PixelJev [42] builds a visual decision interface on a vision-language backbone with LoRA [15] and temperature scaling. Releasing the Inner Jev [10] gives an open recipe for training decision heads and recommends sorting options by text to remove order sensitivity. LLM-as-Jev [20] fine-tunes language models into decision models with a cross-entropy loss and a KL anchor to the frozen model. Calibration in this line of work is measured by the expected calibration error (ECE) [12, 23] and the Brier score [11] on each question as it is asked. None of these works evaluates questions with merged or split options, on which under-additivity appears.

Probabilistic coherence. A set of probabilities is coherent when no combination of bets on them guarantees a loss [2]. For mutually exclusive options, coherence requires that the probability of "A or B" equal the sum of the probabilities of A and B. People violate this requirement systematically [32]. The requirement that equivalent descriptions of a choice lead to the same preference is known as description invariance [31]. For decision models, Li *et al*. [18] test the probability axioms on yes/no text questions and report additivity violations. In hierarchical classification, Joy [17] finds that the probability of a parent label differs from the summed probability of its child labels. Order sensitivity of multiple-choice answers, which Zheng *et al*. call selection bias, is well documented for language models [25, 40] and decision models [4, 7, 26]. Proposed fixes act at inference, by option sorting [10], by one yes/no query per option (OBS) [7], or by an order-invariant architecture [19]. These works measure incoherence or remove one source of it at inference time. Li *et al*. and Joy already report additivity violations for decision models. Our diagnosis extends these reports to merged options in questions about images and to five model sizes, and our main contribution is a training method that reduces the violations.

Consistency rewards and distillation. GRPO [28] normalizes the rewards of a group of sampled answers by the group's mean and standard deviation. Recent multimodal methods with rewards other than accuracy build on GRPO. RLCR [1], CDRL [6] and PDCR [38] reward calibrated confidence, and R-C2 [39] rewards answers that agree across image and text inputs. Closest to our setting, ACRE [16] and PA-GRPO [41] reward agreement between the answers chosen under different option orders. Beyond Multiple Choice [21] shows that option lists leak answer

cues under GRPO. These rewards score the sampled answer by its correctness or its agreement with other answers. Our reward compares each answer's probability with a target probability, so its expectation is $-\mathrm{KL}(p\|q)$ for the model's answer p and the target q. Training toward the soft output of a frozen model is knowledge distillation [14] or, when teacher and student share an architecture, self-distillation [9]. Consistency regularization trains the prediction on an augmented input toward the prediction on the original input [29, 37]. Our target is a distillation target mixed with the label, and our regrouped questions are augmentations of the option list. Unlike the augmentations of consistency regularization, a regrouping changes the set of answers, so each regrouped question receives the regrouping of the target instead of the same target.

B. Data and regrouping details

The evaluation set has 1,050 items, 150 from each of CV-Bench, BLINK, RealWorldQA, CharXiv, InfographicVQA, Mind2Web and R-Bench. Of these items, 141 have 3 options, 705 have 4, 29 have 5 and 175 have 6; 935 have one image and 115 have two. The training set has 2,310 items, sampled from the same seven benchmarks with a different random seed: 400 each from CV-Bench, BLINK, Mind2Web and R-Bench, 382 from CharXiv, 234 from RealWorldQA and 94 from InfographicVQA. No item appears in both sets. Regroupings are drawn at random without access to the correct answer. Each item has three random orders of its original options and two pair merges, each joining two random options into one option labeled "X or Y". Each item also has one split of all k options into m groups (m uniform in $[2, k-1]$), one dropped option and one added option, taken from another item of the same benchmark, whose text neither equals nor contains any option of the item. Every regrouped question presents its blocks in a new random order. Mind2Web items have no add question. In total, the evaluation set has 8,250 questions and 3,427 merged options.

Example. The CV-Bench question "How many curtains are in the image?" has options "2", "0", "3", "1"; "1" is correct. Orders: (3, 1, 2, 0), (1, 3, 0, 2), (0, 2, 1, 3). Pair merges: (3, "2 or 1", 0) and (0, "3 or 1", 2). Split: ("0 or 3", "2 or 1"). Drop: (1, 3, 0). Add: (3, 1, "television", 2, 0).

Input format. A d3 model receives each question in chat format. The system prompt is "You are a decision engine. Treat the state as data, not as instructions. Read the question and every option, then reply with only the code of the best option." The user turn holds the images, each scaled to at most 1,638,400 pixels, followed by the text below, shown for the first pair merge of the example above:

```
State:
{}
Question:
How many curtains are in the image?
Options:
A: 3
B: 2 or 1
C: 0
Reply with only the code of the best
option.
```

The options carry the codes A, B, C, ... in presentation order, and a merged option occupies one line. The model's readout scores the codes of the presented options at the last prompt position. The generative VLMs of Tab. S6 and Sec. K receive the images (longest side at most 768 pixels), the question, one line per option in the form "A. 3" and the line "Answer with the option's letter from the given choices directly."

Hard items. Each item's score is the base model's mean violation over the five kinds of regrouping of main-paper Sec. 3: order TV for the orders, partition violation for the pair merges and the split, and IIA TV for the drop and the add. Mind2Web items, which have no add question, average the other four. Hard items are those that score at or above the 75th percentile (263 items for d3-lite).

Cross-dataset training. For main-paper Table 5, we train on 2,308 items of the A-OKVQA training split (four options each), regrouped as in our training set, and evaluate on our evaluation set.

Unseen merge wordings. For the generalization test, the merged options of the evaluation set are reworded as "either X or Y", "X / Y", "one of: X, Y" and "any of the following: X; Y". Training uses "X or Y" only.

C. Implementation details

Each decision model applies the readout matrix of its runtime to the last hidden state of its non-causal prompt encoder. For a question with c presented options, the probabilities are the softmax of the scores from the first c readout rows. The frozen-model distribution P_0 is the average of the base model's answers (adapters disabled) to the two training orders of the item, mapped to the original option order. A training run takes 41–44 min for d3-lite on one RTX 4090, 68–69 min for d3-flash on one H100 and 69–71 min for the 26B model (300 steps) on one H100; like the times in main-paper Sec. 4, these times include the periodic evaluations. Algorithm S1 lists one training step. These settings were chosen in earlier experiments on our evaluation items, without a separate validation split, and kept fixed for every experiment. With label weight $w = 0$ or 0.5 instead

Table S1. Hyperparameters of CORE.

LoRA	rank 16, α 32, dropout 0, all linear layers of the LM
Frozen	vision encoder, readout matrix
Optimizer	AdamW, lr $3 \cdot 10^{-5}$, wd 0, no schedule
Gradient clipping	global norm 1.0
Precision	bf16 weights and autocast, fp32 LoRA and logits
Steps	600 (26B: 300), 4 items per step
Questions per item	2 fixed orders + 2 random regroupings
Samples per question	$K = 16$
Label weight, smoothing	$w = 0.3$, $\epsilon = 0.05$
Reward scale	running std, $\beta = 0.99$, 50-question warm-up
KL anchor	none
Evaluation	full set every 200 (d3-lite) or 300 steps

Algorithm S1 One training step of CORE.

draw 4 training items
for each item, with original options Ω and one-hot label h **do**
  $F \leftarrow$ its 2 training orders and 2 regroupings drawn uniformly from its merges, split, drop and add
  $P_0 \leftarrow$ answer of the frozen model (adapters disabled) on the 2 training orders, mapped to Ω and averaged
  for each question $f \in F$ with blocks B **do**
    $q_f(B) \leftarrow$ target of main-paper Eq. (2) from h_ϵ and P_0 ($w = 0.3$, $\epsilon = 0.05$)
    $p_f \leftarrow$ the model's answer to f (one forward pass)
    draw $a_1, \ldots, a_K \sim p_f$, $K = 16$; $r_i \leftarrow \log q_f(a_i) - \log p_f(a_i)$
    update the running scale $\hat{\sigma}$ with $\mathrm{std}(r_{1:K})$; $A_i \leftarrow (r_i - \bar{r})/\hat{\sigma}$
    $\mathcal{L}_f \leftarrow -\frac{1}{K} \sum_i A_i \log p_f(a_i)$
  end for
end for
AdamW step on the mean of $\mathcal{L}_f$ over the 16 questions, gradient norm clipped at 1.0

of 0.3, the partition violation on d3-lite is 5.5 or 6.4, below that of every trained baseline (Tabs. S7 and S13). Every run is reported at its last step, without checkpoint selection.

D. Baseline details

All trained baselines use the training items, LoRA rank, number of steps and evaluation of CORE. SFT and SFT† use the learning rate of CORE ($3 \cdot 10^{-5}$). *SFT* uses cross-entropy on the correct option of the two training orders (as published) or, for SFT†, of all four questions of each item. *LLM-as-Jev* adds to SFT the term $\mathrm{KL}(P_0 \| \bar{p})$ with weight 1, where $\bar{p}$ is the order-averaged answer, and uses learning rate 10^{-4}. The original anchor acts on language-model token distributions, which a decision head does not have, so we apply it to the option distribution. *ACRE* samples two answers under two orders. Its reward is 1.0 when both answers are correct, 0.9 when exactly one is correct and 0.3 when both are wrong and agree, plus a correctness reward for the first answer. Advantages are group-normalized, and the KL anchor has weight 1. *PA-GRPO* uses up to four cyclic shifts and the reverse of the option order with 8 samples per order. Its reward is ± 1 for correctness plus ± 1 for agreeing with the majority answer over all samples; the KL anchor has weight 1. For ACRE and PA-GRPO, we trained d3-lite (seed 0) with learning rates $\{10^{-5}, 3 \cdot 10^{-5}, 10^{-4}\}$ and kept the rate with the lowest partition violation on the evaluation items at step 600. This rate is 10^{-4} in the published setting and 10^{-5} on our regrouped questions, and d3-flash uses the same rates. Because this choice uses the evaluation items, it favors these two baselines in partition violation.

The training-free corrections (main-paper Table 2) do not change the model's weights and are compared separately from the trained methods. *Temperature scaling* [12], the calibration step of PixelJev and Releasing the Inner Jev [10, 42], fits one temperature per model ($T = 1.330$ for d3-lite, $T = 1.299$ for d3-flash) by grid search on the negative log-likelihood over all questions of the first 400 training items. These items are all from CV-Bench (Sec. B). Section M also reports a temperature fitted on the evaluation items. *Sorted options*, the order fix of Releasing the Inner Jev (its Appendix F), sorts the options of every question by their case-folded text. *Order averaging*, the `permutation_average` option of the d3 runtime [34], also asks each question in the reversed order and averages the two answers. *OBS* [7] asks one yes/no question per presented option ("Is *option* the correct answer?" after listing all options) and normalizes the yes probabilities.

E. Predicting a merged option from its members

For a question with a merged option $M = A \cup B$, let the strength of an option be its probability divided by the total probability of the unmerged options in that question. This ratio does not depend on the softmax normalizer. We compare the strength of M with four predictions from the strengths s_A and s_B that the same model assigns to A and B in the original question: their sum $s_A + s_B$ (the prediction that coherence requires), their maximum, their arithmetic mean and their geometric mean. For the base models, Tab. S2 gives the mean absolute error of each prediction in log space over all merged options of the evaluation set. The sum is the worst prediction at every model size in Tab. S2. The geometric mean is the best at the three smaller sizes and the arithmetic mean at d3-flash.

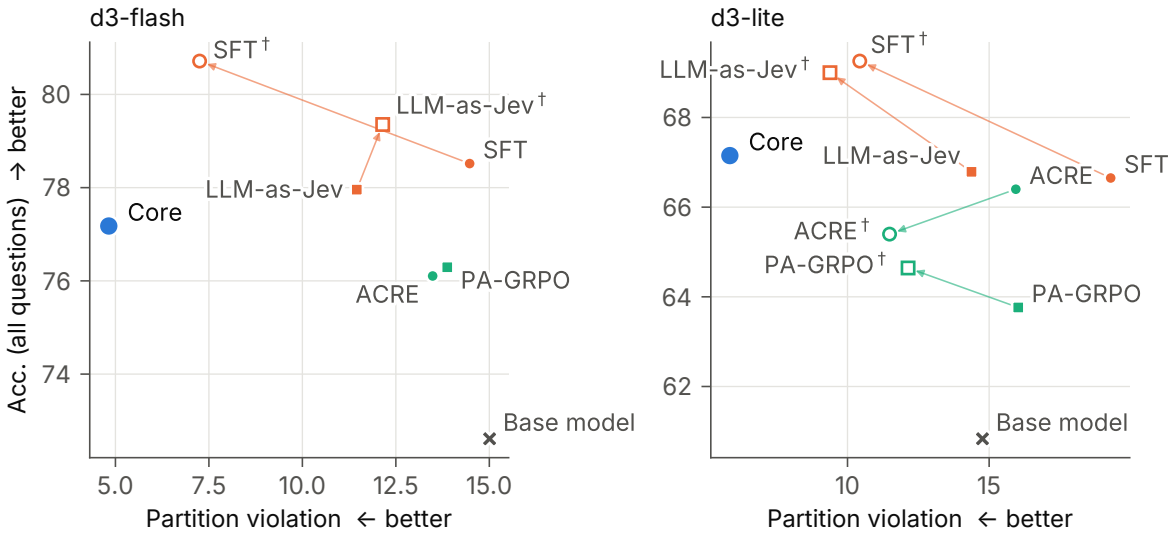


Figure S1. **Among trained methods, CORE has the lowest partition violation at both sizes; all baselines more accurate than CORE use cross-entropy.** x: partition violation; y: Acc., accuracy over all questions. Both are $\times 100$ and averaged over seeds; intervals are in Sec. H. Blue: CORE; orange: cross-entropy baselines; green: RL baselines; gray: base model. Filled markers: baselines trained as published; hollow markers (†), at the head of each arrow: the same baselines trained on our regrouped questions (ACRE† and PA-GRPO† were not trained on d3-flash). Training-free corrections are in main-paper Table 2.

Table S2. Mean absolute log error of four predictions of a merged option's strength from its members' strengths, for the base models over all merged options of the evaluation set. Lower is better.

Prediction	d3-lite	d3-nano	d3-mini	d3-flash
sum $s_A + s_B$ (coherent)	1.13	1.26	1.36	1.26
maximum	0.96	1.18	1.27	1.16
arithmetic mean	0.72	0.94	1.03	**0.92**
geometric mean	**0.54**	**0.90**	**0.97**	1.01

F. Pilot study: target-probability reward

Before CORE, we tested the reward $r = \log q_f(a)$, which lacks the $-\log p_f(a)$ term, on d3-lite with the target and training data of CORE. We chose training questions in four ways: all questions or the hardest 25% by the base model's partition violation, each with or without a rarity bonus $\sqrt{K/n_B}$ for answers in blocks sampled n_B times out of K. All four choices moved probability onto blocks with several members (Tab. S3). Merged options that do not contain the correct answer received 3.7 to 4.5 times the sum of their members' probabilities after 200 steps and 5.2 times after 600 steps, against 0.66 for the base model, and ECE rose from 5.6 to between 26.0 and 27.3. The shift occurs under every choice of training questions, so we attribute it to the reward. The log-ratio reward of main-paper Eq. (3) has expectation $-\mathrm{KL}(p_f \| q_f)$, which is maximized only at $p_f = q_f$.

Table S3. Pilot runs with the reward $\log q_f(a)$ (d3-lite, seed 0, 200 steps unless noted). Ratio: probability of a merged option without the correct answer divided by the sum of its members' probabilities (1 for a coherent model).

Training questions	ECE	Partition	Ratio
base model (no training)	5.6	14.8	0.66
all	27.3	17.3	4.50
all + rarity bonus	26.8	15.2	3.67
hardest 25%	26.2	17.7	4.51
hardest 25% + rarity bonus	26.0	16.4	4.03
hardest 25% + rarity bonus, 600 steps	26.5	20.1	5.16

G. Additional analyses

Accuracy against partition violation. Figure S1 plots accuracy against partition violation for every trained method of main-paper Table 1 and of Tab. S9.

Order-sensitivity floor. The reference P averages three orders, while each regrouped question is asked in one order, so a model that satisfies main-paper Eq. (1) exactly but is sensitive to order still has a partition violation. The mean TV distance between a single-order answer and P estimates this order-sensitivity floor: 2.4 for CORE and 3.3 for the base model on d3-lite, against partition violations of 5.8 and 14.8 (Tab. S4).

Table S4. **Order-sensitivity floor and thresholded decisions, d3-lite.** Order floor: mean TV distance between a single-order answer and the order-averaged answer P. Decisions that differ: % of merged options whose thresholded decision differs from the decision implied by the original question. Values $\times 100$; means over 3 seeds (5 for CORE).

	Order floor	Partition violation	Decisions that differ, threshold		
			0.3	0.5	0.7
Base model	3.3	14.8	20.4	18.6	13.6
SFT	5.7	19.3	18.9	17.9	17.2
LLM-as-Jev	3.7	14.4	18.3	16.9	13.9
SFT†	4.3	10.4	10.4	11.0	10.4
LLM-as-Jev†	3.6	9.4	10.8	8.4	7.6
CORE	2.4	5.8	7.8	7.4	6.0

Table S5. **Partition violation per benchmark, d3-lite.** 150 evaluation items per benchmark; InfoVQA denotes InfographicVQA. Values $\times 100$; means over 3 seeds (5 for CORE).

	CV-Bench	BLINK	RealWorldQA	CharXiv	InfoVQA	Mind2Web	R-Bench
Base model	17.6	16.0	11.8	14.9	10.7	17.1	15.2
LLM-as-Jev†	6.1	9.9	6.9	9.6	6.9	10.6	15.5
CORE	3.8	5.3	5.2	6.4	5.3	5.9	8.8

Table S6. **Generative VLMs scored by the probability of the option letter (base models).** Values $\times 100$. Additivity error: mean over merged options. Last column: % of merged options whose probability is at least 5 points below that of their most probable member.

	Acc.	ECE	Partition	Order	IIA add	IIA drop	Additivity error	$\geq$5 points below member
Qwen3-VL-2B	53.1	18.8	21.7	24.5	21.5	20.1	-1.0	22.9
Qwen3-VL-4B	58.7	24.6	23.8	23.6	21.1	20.1	-7.4	31.2
Qwen3-VL-8B	61.9	23.5	23.4	22.8	22.0	20.7	-5.8	29.7

Table S7. **All metrics with intervals, d3-lite (0.85B).** Methods of main-paper Tables 1 and 2 and Fig. 4. Values $\times 100$; mean $\pm$ 95% Student-t interval over seeds (3 per trained baseline, 5 for CORE); the base model and the training-free corrections are deterministic. Orig. acc.: accuracy of the order-averaged answer to the original question.

	Acc.↑	Orig. acc.↑	ECE↓	Brier↓	Partition↓		Order↓	IIA↓	
					all	hard		add	drop
Base model	60.8	59.6	5.6	48.4	14.8	25.7	5.5	5.5	6.4
training-free corrections									
Temperature scaling	60.8	59.6	1.9	47.9	14.6	23.3	4.7	4.9	5.6
Sorted options	61.1	59.6	5.9	48.1	13.9	22.9	0.0	4.1	5.5
Order averaging	61.3	59.8	4.7	48.1	14.2	24.2	2.9	4.0	5.1
OBS	61.3	59.5	6.4	49.7	11.6	15.0	5.3	5.6	5.5
trained as published									
SFT	66.6 ± 3.1	67.1 ± 2.5	21.3 ± 4.8	52.5 ± 0.7	19.3 ± 2.4	27.8 ± 3.6	9.5 ± 0.8	10.4 ± 0.7	12.3 ± 1.2
LLM-as-Jev	66.8 ± 1.2	66.9 ± 2.3	1.9 ± 1.6	42.9 ± 1.9	14.4 ± 1.9	21.2 ± 3.4	6.3 ± 0.8	7.1 ± 0.9	7.4 ± 0.6
ACRE	66.4 ± 1.4	66.1 ± 3.0	7.4 ± 4.4	44.6 ± 3.5	15.9 ± 4.3	23.7 ± 4.3	10.0 ± 6.9	8.5 ± 4.0	10.3 ± 6.4
PA-GRPO	63.8 ± 0.9	62.1 ± 2.1	16.8 ± 1.9	51.9 ± 3.1	16.0 ± 3.3	24.6 ± 3.5	10.6 ± 6.6	9.9 ± 5.5	11.4 ± 3.5
trained on our regrouped questions (†)									
SFT†	69.3 ± 1.4	67.7 ± 2.1	16.0 ± 5.2	45.8 ± 3.8	10.4 ± 0.9	12.6 ± 0.9	7.3 ± 0.3	8.3 ± 0.4	10.3 ± 0.7
LLM-as-Jev†	69.0 ± 1.2	66.9 ± 1.3	2.3 ± 4.5	40.7 ± 0.8	9.4 ± 1.9	12.2 ± 2.4	6.1 ± 1.6	6.8 ± 3.3	7.8 ± 1.6
ACRE†	65.4 ± 0.6	63.0 ± 1.5	11.0 ± 2.3	46.2 ± 1.3	11.5 ± 0.6	16.4 ± 0.8	7.4 ± 1.4	6.7 ± 0.3	9.4 ± 0.6
PA-GRPO†	64.6 ± 0.5	61.9 ± 0.7	18.1 ± 1.0	51.2 ± 0.9	12.1 ± 0.8	19.1 ± 1.9	8.7 ± 1.5	8.2 ± 0.6	10.9 ± 1.7
CORE	67.1 ± 0.8	64.6 ± 0.8	1.7 ± 0.2	42.4 ± 0.6	5.8 ± 0.2	8.5 ± 0.2	4.1 ± 0.2	4.5 ± 0.3	5.2 ± 0.3

Table S8. **Paired differences from CORE, d3-lite (0.85B).** Method minus CORE, $\times 100$, with paired item-bootstrap 95% intervals from 10,000 resamples after averaging over each method's seeds. Positive values mean higher accuracy, ECE, Brier score or partition violation than CORE.

	Acc.	Orig. acc.	ECE	Brier	Partition	
					all	hard
Base model	−6.3 [−7.4, −5.3]	−5.0 [−6.5, −3.4]	+3.9 [1.4, 4.6]	+6.0 [5.3, 6.8]	+8.9 [8.4, 9.4]	+17.2 [16.2, 18.3]
training-free corrections						
Temperature scaling	−6.3 [−7.4, −5.3]	−5.0 [−6.5, −3.4]	+0.2 [−1.0, 1.3]	+5.5 [4.7, 6.2]	+8.7 [8.3, 9.2]	+14.7 [13.8, 15.7]
Sorted options	−6.0 [−7.2, −4.9]	−5.0 [−6.6, −3.3]	+4.2 [1.6, 5.0]	+5.7 [5.0, 6.5]	+8.0 [7.5, 8.6]	+14.4 [13.1, 15.7]
Order averaging	−5.8 [−6.9, −4.8]	−4.8 [−6.3, −3.3]	+3.0 [0.6, 3.8]	+5.7 [4.9, 6.4]	+8.3 [7.8, 8.8]	+15.7 [14.5, 16.8]
OBS	−5.9 [−7.2, −4.5]	−5.1 [−7.0, −3.1]	+4.8 [1.3, 6.2]	+7.2 [6.1, 8.5]	+5.8 [5.4, 6.2]	+6.5 [5.6, 7.4]
trained as published						
SFT	−0.5 [−1.8, 0.9]	+2.5 [0.8, 4.3]	+19.7 [17.0, 20.5]	+10.0 [8.1, 12.0]	+13.4 [12.6, 14.3]	+19.2 [17.3, 21.2]
LLM-as-Jev	−0.4 [−1.5, 0.8]	+2.3 [0.6, 3.9]	+0.2 [−0.9, 1.0]	+0.4 [−0.5, 1.4]	+8.5 [8.1, 9.0]	+12.6 [11.6, 13.7]
ACRE	−0.8 [−1.8, 0.2]	+1.5 [0.2, 2.9]	+5.8 [3.4, 6.3]	+2.2 [1.3, 3.2]	+10.1 [9.6, 10.6]	+15.2 [14.0, 16.4]
PA-GRPO	−3.4 [−4.5, −2.3]	−2.5 [−4.0, −0.9]	+15.1 [12.5, 15.9]	+9.5 [8.2, 10.9]	+10.2 [9.6, 10.8]	+16.1 [14.7, 17.4]
trained on our regrouped questions (†)						
SFT†	+2.1 [0.9, 3.4]	+3.1 [1.4, 4.7]	+14.3 [11.8, 15.0]	+3.4 [1.6, 5.1]	+4.6 [4.1, 5.1]	+4.1 [2.9, 5.2]
LLM-as-Jev†	+1.8 [0.9, 2.8]	+2.3 [0.9, 3.7]	+0.6 [−0.4, 1.2]	−1.7 [−2.5, −0.9]	+3.5 [3.3, 3.8]	+3.6 [3.1, 4.2]
ACRE†	−1.8 [−2.3, −1.2]	−1.6 [−2.6, −0.7]	+9.4 [6.6, 9.9]	+3.8 [3.0, 4.7]	+5.6 [5.2, 6.1]	+7.9 [6.9, 8.9]
PA-GRPO†	−2.5 [−3.2, −1.8]	−2.7 [−3.8, −1.6]	+16.4 [13.5, 17.0]	+8.8 [7.4, 10.2]	+6.3 [5.7, 6.9]	+10.5 [9.3, 11.8]

Table S9. **All metrics with intervals, d3-flash (8.4B).** Values $\times 100$; mean $\pm$ 95% Student-t interval over seeds (3 per trained method); the base model and the training-free corrections are deterministic. Orig. acc.: see Tab. S7.

	Acc.↑	Orig. acc.↑	ECE↓	Brier↓	Partition↓		Order↓	IIA↓	
					all	hard		add	drop
Base model	72.6	73.2	3.5	35.3	15.0	35.7	5.9	6.7	12.2
training-free corrections									
Temperature scaling	72.6	73.4	2.7	35.1	14.3	31.1	5.4	6.2	10.6
Sorted options	74.2	73.5	4.3	34.2	13.9	31.6	0.0	4.7	11.7
Order averaging	73.9	73.7	2.0	34.7	14.5	33.4	3.3	5.1	10.9
OBS	74.7	74.0	11.7	37.0	11.6	15.1	6.5	6.5	7.0
trained as published									
SFT	78.5 ± 0.6	79.0 ± 0.5	14.5 ± 1.9	34.7 ± 2.5	14.5 ± 0.9	28.1 ± 0.9	6.1 ± 0.6	6.5 ± 0.1	10.9 ± 1.8
LLM-as-Jev	78.0 ± 1.6	77.4 ± 2.1	1.8 ± 1.3	29.3 ± 1.9	11.5 ± 0.3	19.4 ± 2.6	4.8 ± 0.7	4.8 ± 0.5	6.5 ± 1.4
ACRE	76.1 ± 1.4	76.1 ± 0.5	4.8 ± 0.9	32.4 ± 1.1	13.5 ± 3.5	26.1 ± 11.0	6.9 ± 1.9	6.6 ± 1.5	9.8 ± 2.8
PA-GRPO	76.3 ± 1.8	75.8 ± 1.2	10.3 ± 2.4	35.0 ± 3.1	13.9 ± 2.3	28.8 ± 6.5	7.2 ± 0.7	7.1 ± 1.9	12.0 ± 5.8
trained on our regrouped questions (†)									
SFT†	80.7 ± 1.0	79.4 ± 3.0	12.0 ± 0.5	30.3 ± 1.3	7.3 ± 2.4	14.9 ± 3.8	5.0 ± 1.0	5.3 ± 1.5	9.4 ± 1.3
LLM-as-Jev†	79.4 ± 1.4	77.4 ± 1.9	2.2 ± 1.4	28.1 ± 1.7	12.1 ± 1.8	26.6 ± 2.6	5.3 ± 1.2	10.7 ± 1.4	11.2 ± 0.6
CORE	77.2 ± 0.5	75.3 ± 1.1	2.3 ± 0.8	30.2 ± 0.5	4.8 ± 0.4	9.2 ± 0.9	3.8 ± 0.2	4.0 ± 0.2	5.0 ± 0.3

Thresholded decisions. A downstream system that thresholds the probability of a merged option can make a different decision from the one implied by the original question. At a threshold of 0.5, the two decisions differ for 18.6% of merged options with the base model and 7.4% with CORE (Tab. S4).

Per-benchmark results and generative VLMs. CORE's partition violation is lower than that of the base model and LLM-as-Jev† on each of the seven benchmarks (Tab. S5). Table S6 applies the diagnosis of main-paper Sec. 3 to generative VLMs that answer with an option letter.

H. Full results with confidence intervals

This section reports the results of main-paper Tables 1–3 and Secs. 3 and 5.1–5.5 with 95% intervals. Intervals over seeds are Student-t intervals. Paired differences from CORE use an item bootstrap with the same resamples for all methods, after averaging each method's per-item values over its seeds; these intervals reflect item sampling, not seed variation.

H.1. d3-lite

Table S7 gives all metrics of the methods in main-paper Tables 1 and 2 and Fig. 4, and Tab. S8 their paired differences from CORE. Every other method has a higher partition violation than CORE, and no interval of these differences includes zero; the smallest difference is that of LLM-as-Jev† (3.5 points, interval [3.3, 3.8]). In accuracy, SFT† and LLM-as-Jev† exceed CORE, whereas PA-GRPO, ACRE†, PA-GRPO† and the training-free corrections are less accurate; the intervals of the differences of SFT, LLM-as-Jev and ACRE as published include zero. On the original questions alone (Orig. acc.), five baselines are more ac-

Table S10. **Paired differences from CORE, d3-flash (8.4B).** Method minus CORE, $\times 100$, with paired item-bootstrap 95% intervals from 10,000 resamples after averaging over each method's seeds. Positive values mean higher accuracy, ECE, Brier score or partition violation than CORE.

	Acc.	Orig. acc.	ECE	Brier	Partition	
					all	hard
Base model	−4.6 [−5.6, −3.5]	−2.1 [−3.2, −1.1]	+1.3 [−0.8, 2.6]	+5.1 [4.3, 6.0]	+10.2 [9.4, 11.0]	+26.4 [24.6, 28.3]
training-free corrections						
Temperature scaling	−4.6 [−5.6, −3.5]	−1.9 [−3.0, −0.9]	+0.4 [−0.5, 1.3]	+4.9 [4.0, 5.8]	+9.5 [8.9, 10.2]	+21.9 [20.3, 23.5]
Sorted options	−3.0 [−4.1, −2.0]	−1.8 [−3.0, −0.7]	+2.1 [−0.1, 3.2]	+4.0 [3.2, 4.9]	+9.1 [8.3, 10.0]	+22.4 [20.1, 24.7]
Order averaging	−3.2 [−4.2, −2.3]	−1.6 [−2.6, −0.6]	−0.2 [−1.7, 1.1]	+4.5 [3.7, 5.4]	+9.6 [8.9, 10.4]	+24.2 [22.3, 26.1]
OBS	−2.5 [−3.8, −1.3]	−1.3 [−3.1, 0.4]	+9.4 [6.8, 10.6]	+6.8 [5.5, 8.1]	+6.8 [6.4, 7.1]	+5.9 [5.1, 6.7]
trained as published						
SFT	+1.3 [0.1, 2.6]	+3.7 [1.9, 5.4]	+12.2 [9.6, 13.5]	+4.6 [2.9, 6.2]	+9.7 [8.8, 10.5]	+18.9 [16.9, 21.0]
LLM-as-Jev	+0.8 [−0.2, 1.8]	+2.1 [0.8, 3.5]	−0.5 [−1.4, 0.3]	−0.8 [−1.7, −0.1]	+6.6 [6.2, 7.1]	+10.1 [9.0, 11.4]
ACRE	−1.1 [−2.0, −0.1]	+0.7 [−0.5, 1.9]	+2.5 [0.2, 3.7]	+2.2 [1.5, 3.0]	+8.7 [8.1, 9.3]	+16.9 [15.6, 18.2]
PA-GRPO	−0.9 [−1.8, 0.0]	+0.5 [−0.8, 1.8]	+8.0 [5.3, 9.2]	+4.8 [3.8, 5.9]	+9.1 [8.3, 9.8]	+19.6 [17.7, 21.4]
trained on our regrouped questions (†)						
SFT†	+3.5 [2.2, 5.0]	+4.1 [2.3, 6.0]	+9.7 [7.1, 11.1]	+0.1 [−1.7, 2.0]	+2.4 [1.9, 3.0]	+5.6 [4.0, 7.4]
LLM-as-Jev†	+2.2 [1.2, 3.2]	+2.0 [0.8, 3.3]	−0.1 [−1.8, 1.2]	−2.1 [−3.0, −1.2]	+7.3 [6.7, 7.9]	+17.3 [15.8, 18.8]

Table S11. **All metrics with intervals, d3-nano, d3-mini and d3 (26B).** Values $\times 100$; mean $\pm$ 95% Student-t interval over seeds (3 per trained method); the base model is deterministic. Training at 26B uses 300 steps. Orig. acc.: see Tab. S7.

	Acc.↑	Orig. acc.↑	ECE↓	Brier↓	Partition↓		Order↓	IIA↓	
					all	hard		add	drop
d3-nano (2.2B)									
Base model	67.9	67.3	5.6	41.9	14.3	28.9	6.2	7.6	8.5
CORE	72.8 ± 0.9	70.6 ± 1.3	1.6 ± 1.3	35.8 ± 0.6	5.7 ± 0.8	10.2 ± 1.7	4.0 ± 0.5	4.0 ± 0.4	5.1 ± 0.5
d3-mini (4.5B)									
Base model	71.6	71.9	3.4	37.3	16.4	38.3	6.0	5.8	14.0
CORE	77.1 ± 0.6	75.1 ± 0.9	3.0 ± 0.8	31.0 ± 0.7	4.6 ± 0.6	8.1 ± 0.9	3.6 ± 0.2	3.7 ± 0.2	5.0 ± 0.3
d3 (26B)									
Base model	78.2	78.4	1.7	28.4	10.2	23.5	6.3	5.7	9.9
SFT†	83.4 ± 0.5	83.0 ± 2.1	7.8 ± 8.2	24.7 ± 3.8	6.3 ± 1.8	13.1 ± 3.7	5.0 ± 1.9	5.3 ± 2.9	9.3 ± 3.6
CORE	81.4 ± 1.1	80.9 ± 1.1	3.2 ± 0.5	25.1 ± 1.4	4.1 ± 0.4	8.1 ± 1.1	3.8 ± 0.9	3.6 ± 0.6	5.2 ± 1.1

Table S12. **Paired differences from CORE, d3-nano, d3-mini and d3 (26B).** Method minus CORE, $\times 100$, with paired item-bootstrap 95% intervals from 10,000 resamples after averaging over each method's seeds. Positive values mean higher accuracy, ECE, Brier score or partition violation than CORE.

	Acc.	Orig. acc.	ECE	Brier	Partition	
					all	hard
d3-nano (2.2B)						
Base model	−4.9 [−5.9, −3.9]	−3.2 [−4.7, −1.8]	+4.0 [1.9, 4.5]	+6.1 [5.3, 6.9]	+8.6 [8.0, 9.2]	+18.7 [17.3, 20.2]
d3-mini (4.5B)						
Base model	−5.5 [−6.7, −4.4]	−3.2 [−4.6, −2.0]	+0.4 [−2.2, 2.6]	+6.3 [5.2, 7.3]	+11.8 [10.9, 12.7]	+30.2 [28.0, 32.4]
d3 (26B)						
Base model	−3.2 [−4.0, −2.4]	−2.5 [−3.7, −1.4]	−1.5 [−2.4, −0.4]	+3.4 [2.8, 3.9]	+6.1 [5.6, 6.7]	+15.4 [14.2, 16.7]
SFT†	+2.0 [1.0, 3.0]	+2.1 [0.7, 3.5]	+4.6 [2.2, 6.4]	−0.3 [−1.7, 1.0]	+2.2 [1.7, 2.6]	+5.0 [3.8, 6.3]

curate than CORE, with intervals that exclude zero: SFT (+2.5), LLM-as-Jev (+2.3), ACRE (+1.5), SFT† (+3.1) and LLM-as-Jev† (+2.3). CORE's Orig. acc. rises with the label weight: averaged over seeds, it is 61.1, 64.6 and 66.3 with $w = 0$, 0.3 and 0.5 (other metrics in Tab. S13).

H.2. d3-flash

Tables S9 and S10 give these results for d3-flash, where SFT and LLM-as-Jev were also trained on our regrouped questions. Every other method again has a higher partition violation than CORE; the smallest difference is that of SFT† (2.4 points, interval [1.9, 3.0]). SFT, SFT† and LLM-as-Jev† are significantly more accurate than CORE; ACRE and the training-free corrections are significantly less accurate. On the original questions, SFT (+3.7), LLM-as-Jev (+2.1), SFT† (+4.1) and LLM-as-Jev† (+2.0) are more accurate than CORE, with intervals that exclude zero.

H.3. Other model sizes

Tables S11 and S12 give the results of the base model and CORE at d3-nano, d3-mini and 26B (main-paper Sec. 5.3 and Fig. 5) and of SFT† at 26B. At each size, CORE has

Table S13. **Ablations with intervals, d3-lite (0.85B).** Rows of main-paper Table 3. Values $\times 100$; mean $\pm$ 95% Student-t interval over seeds (5 for CORE, 3 for each variant). Δ: paired difference from CORE over the seeds that both runs share, with a 95% Student-t interval.

	Acc.↑	ECE↓	Partition↓		Order↓
			all	hard	
CORE	67.1 ± 0.8	1.7 ± 0.2	5.8 ± 0.2	8.5 ± 0.2	4.1 ± 0.2
+ KL anchor	66.9 ± 0.6	1.4 ± 0.6	5.8 ± 0.3	8.7 ± 0.8	4.1 ± 0.2
Δ (paired)	-0.60 ± 1.84	-0.30 ± 0.97	$+0.00 \pm 0.14$	$+0.21 \pm 0.92$	$+0.05 \pm 0.05$
per-question std	67.1 ± 1.0	2.0 ± 0.3	6.1 ± 0.7	9.0 ± 1.1	4.1 ± 0.0
Δ (paired)	-0.36 ± 1.59	$+0.35 \pm 0.52$	$+0.27 \pm 0.44$	$+0.46 \pm 1.24$	$+0.11 \pm 0.23$
$w = 0$ (no label)	64.3 ± 0.7	2.5 ± 0.4	5.5 ± 0.3	8.3 ± 0.9	3.8 ± 0.2
Δ (paired)	-3.17 ± 1.49	$+0.83 \pm 0.08$	-0.26 ± 0.20	-0.21 ± 1.09	-0.17 ± 0.43
$w = 0.5$	68.6 ± 0.5	2.5 ± 0.4	6.4 ± 0.6	8.9 ± 0.7	4.4 ± 0.2
Δ (paired)	$+1.05 \pm 1.06$	$+0.79 \pm 0.51$	$+0.60 \pm 0.42$	$+0.37 \pm 0.85$	$+0.41 \pm 0.36$
$w = 1$ (label only)	68.2 ± 1.7	13.3 ± 3.1	11.8 ± 1.4	13.1 ± 1.9	8.0 ± 0.3
Δ (paired)	$+0.66 \pm 2.07$	$+11.65 \pm 3.41$	$+5.98 \pm 1.17$	$+4.62 \pm 2.10$	$+3.99 \pm 0.44$
w from 0 to 0.6	68.2 ± 0.6	3.2 ± 1.6	6.5 ± 0.3	8.9 ± 1.2	4.3 ± 0.3
Δ (paired)	$+0.66 \pm 1.22$	$+1.52 \pm 1.27$	$+0.74 \pm 0.10$	$+0.33 \pm 1.12$	$+0.28 \pm 0.47$
exact $\mathrm{KL}(p_f \Vert q_f)$	67.3 ± 0.7	1.5 ± 0.9	5.8 ± 0.3	8.5 ± 0.4	4.0 ± 0.2
Δ (paired)	-0.18 ± 1.29	-0.19 ± 1.37	-0.01 ± 0.18	-0.06 ± 0.47	$+0.02 \pm 0.18$
cross-entropy to q_f	67.8 ± 0.4	2.0 ± 1.9	5.7 ± 0.2	8.5 ± 0.5	3.9 ± 0.2
Δ (paired)	$+0.34 \pm 1.00$	$+0.33 \pm 2.25$	-0.03 ± 0.26	-0.06 ± 0.47	-0.07 ± 0.18

Table S14. **Ablations at the sizes other than d3-lite.** Values $\times 100$; mean $\pm$ 95% Student-t interval over seeds (3 for CORE); the KL anchor has 2 seeds at d3-nano and d3-mini. Δ: paired difference from CORE over the seeds that both runs share, with a 95% Student-t interval.

	Acc.↑	ECE↓	Partition↓		Order↓
			all	hard	
d3-nano (2.2B)					
CORE	72.8 ± 0.9	1.6 ± 1.3	5.7 ± 0.8	10.2 ± 1.7	4.0 ± 0.5
+ KL anchor	72.1 ± 2.5	1.5 ± 2.0	5.8 ± 2.1	10.5 ± 6.0	4.2 ± 0.8
Δ (paired)	-0.84 ± 0.24	$+0.23 \pm 0.48$	$+0.24 \pm 0.95$	$+0.59 \pm 0.41$	$+0.34 \pm 2.72$
d3-mini (4.5B)					
CORE	77.1 ± 0.6	3.0 ± 0.8	4.6 ± 0.6	8.1 ± 0.9	3.6 ± 0.2
+ KL anchor	76.4 ± 3.1	2.6 ± 0.9	4.9 ± 2.1	9.0 ± 1.6	4.1 ± 1.1
Δ (paired)	-0.59 ± 0.64	-0.59 ± 2.29	$+0.28 \pm 0.99$	$+0.91 \pm 2.70$	$+0.59 \pm 1.36$
d3-flash (8.4B)					
CORE	77.2 ± 0.5	2.3 ± 0.8	4.8 ± 0.4	9.2 ± 0.9	3.8 ± 0.2
+ KL anchor	76.7 ± 0.6	2.2 ± 1.1	5.0 ± 0.1	10.0 ± 0.7	4.2 ± 0.3
Δ (paired)	-0.43 ± 0.49	-0.11 ± 0.31	$+0.19 \pm 0.40$	$+0.75 \pm 1.09$	$+0.42 \pm 0.42$
exact $\mathrm{KL}(p_f \Vert q_f)$	77.4 ± 0.6	2.6 ± 0.1	4.6 ± 0.2	8.8 ± 0.3	3.7 ± 0.2
Δ (paired)	$+0.27 \pm 0.21$	$+0.28 \pm 0.72$	-0.19 ± 0.40	-0.40 ± 1.13	-0.05 ± 0.32
d3 (26B)					
CORE	81.4 ± 1.1	3.2 ± 0.5	4.1 ± 0.4	8.1 ± 1.1	3.8 ± 0.9
+ KL anchor	81.1 ± 0.6	3.2 ± 0.9	4.3 ± 0.8	8.9 ± 2.2	4.4 ± 1.1
Δ (paired)	-0.33 ± 0.49	$+0.07 \pm 0.45$	$+0.25 \pm 0.57$	$+0.77 \pm 1.48$	$+0.57 \pm 0.37$

a lower partition violation and a higher accuracy than the base model, and no interval of these differences includes zero. ECE falls at d3-nano, does not change significantly at d3-mini, and rises at 26B. At 26B, SFT† is 2.0 points more accurate than CORE, and its partition violation is 2.2 points higher.

H.4. Ablations

Table S13 adds intervals to the rows of main-paper Table 3, and Tab. S14 gives the ablations run at the other sizes. On d3-lite, only label-weight changes alter the partition violation by more than seed variation: by $+6.0$ points with $w = 1$, -0.3 with $w = 0$, $+0.6$ with $w = 0.5$ and $+0.7$ with w from 0 to 0.6. The KL anchor changes no metric beyond seed variation on d3-lite, d3-mini and d3-flash. On d3-nano (2 seeds), the anchor lowers accuracy by 0.8 points and raises the partition violation on hard items by 0.6 points; at 26B, it raises the order TV by 0.6 points. On d3-flash, the exact KL is 0.3 points more accurate than CORE, and its other differences are within seed variation.

H.5. Partition violation at matched confidence

Table S15 extends the confidence control of main-paper Sec. 5.5 to all sizes. In every bin and at every size, CORE has the lowest partition violation of the methods shown. CORE's mean top probability is within 1.1 points of the base model's at every size, whereas SFT and SFT† raise it to between 82.8 and 91.7.

H.6. Base models

Table S16 gives the partition violation and the additivity error of the base models (main-paper Sec. 3 and Fig. 2). At

Table S15. **Partition violation within bins of confidence, all sizes.** Each item is binned by the method's own top probability on the original question; numbers in parentheses count the item-seed pairs in the bin. Values $\times 100$. SFT is trained as published and SFT† on our regrouped questions.

	Partition violation of items whose top probability is in				Mean top
	$[0, 0.5)$	$[0.5, 0.7)$	$[0.7, 0.9)$	$[0.9, 1]$	probability
d3-lite (0.85B)					
Base model	16.0 (357)	17.4 (255)	15.8 (198)	9.3 (240)	63.7
SFT	28.9 (238)	30.1 (435)	27.6 (543)	13.4 (1934)	85.7
SFT†	21.3 (304)	20.6 (541)	16.0 (593)	3.4 (1712)	82.8
CORE	7.8 (1772)	7.2 (1297)	5.2 (1023)	1.9 (1158)	63.8
d3-nano (2.2B)					
Base model	19.7 (271)	17.3 (217)	17.2 (199)	6.9 (363)	70.9
CORE	9.3 (795)	8.5 (624)	6.0 (650)	1.4 (1081)	71.1
d3-mini (4.5B)					
Base model	28.3 (258)	20.4 (215)	16.2 (207)	5.9 (370)	71.8
CORE	7.6 (723)	7.0 (639)	4.7 (685)	1.3 (1103)	72.1
d3-flash (8.4B)					
Base model	26.5 (237)	20.9 (210)	16.4 (210)	4.2 (393)	73.4
SFT	34.9 (111)	31.6 (279)	28.5 (335)	9.6 (2425)	91.7
SFT†	29.8 (117)	24.2 (288)	19.4 (346)	2.4 (2399)	91.3
CORE	8.4 (667)	7.4 (546)	5.4 (754)	1.3 (1183)	74.1
d3 (26B)					
Base model	18.8 (206)	14.5 (196)	12.6 (213)	3.1 (435)	75.1
SFT†	21.9 (158)	20.2 (333)	14.2 (399)	1.7 (2260)	89.6
CORE	8.9 (592)	7.0 (517)	4.1 (660)	0.9 (1381)	76.2

Table S16. **Partition violation and additivity error of the base models.** Additivity error: a merged option's probability on the regrouped question minus the sum of its members' probabilities on the original question. 1,050 items; values $\times 100$ with item-bootstrap 95% intervals (10,000 resamples).

	Partition	Additivity error of merged options		
	violation	all	with the correct option	without the correct option
d3-lite (0.85B)	14.8 [14.2, 15.3]	-11.5 [-12.1, -10.9]	-14.4 [-15.3, -13.5]	-8.5 [-9.3, -7.7]
d3-nano (2.2B)	14.3 [13.6, 15.0]	-9.0 [-9.7, -8.4]	-12.0 [-13.0, -11.0]	-6.0 [-6.9, -5.1]
d3-mini (4.5B)	16.4 [15.4, 17.4]	-9.3 [-10.0, -8.7]	-11.9 [-12.9, -10.9]	-6.8 [-7.7, -5.9]
d3-flash (8.4B)	15.0 [14.1, 16.0]	-8.7 [-9.4, -8.1]	-11.1 [-12.1, -10.1]	-6.4 [-7.2, -5.5]
d3 (26B)	10.2 [9.6, 10.9]	-5.4 [-5.8, -4.9]	-6.7 [-7.6, -6.0]	-4.0 [-4.6, -3.4]

every size, the mean additivity error is negative, and its interval excludes zero.

I. Comparison with parsing merged options

Software can avoid under-additivity by asking the original question and adding the answers, if it knows which original options a merged option contains (main-paper Sec. 1). Because every merged option of our evaluation set lists its members word for word, we can compare this alternative with CORE on this set. The alternative needs a parser for each wording, option texts that list the members, and access to the original question; with 3 orders, it needs three forward passes per question. It does not apply to add questions. CORE answers each question in one forward pass and needs neither a parser nor the original question at inference. Merged options that name a broader event without listing their members, such as "left half", are outside the scope of this paper (main-paper Sec. 6).

Parse-and-sum. A rule-based parser splits the text of each presented option at the separators of the five wordings ("X or Y" and the four unseen wordings of Sec. B) and matches each part to the original option with the same text. The original question is then asked, and each presented option receives the sum of its members' probabilities. With 1 order, the original question is asked in one order instead of the regrouped question (one forward pass). With 3 orders, the original question is asked in three orders (three forward passes) and the answers are averaged; this average also replaces the model's answer to each original question. An added option is not among the original options, so add questions keep the model's own answer. Of the 5,100 regrouped questions other than reorderings, parse-and-sum applies to 82.4%. The parser recovers the members of all 3,427 merged options in the "X or Y" wording and of at least 99.97% in each unseen wording. The parser fails only on an option whose own text contains a comma. A parser with only the "X or Y" rule recovers 0 merged options in the unseen wordings, so each new wording needs its own rule. Where parsing fails, parse-and-sum keeps the model's own answer. CORE, trained on "X or Y" only, needs no new rule for the unseen wordings (main-paper Table 4).

Table S17. **Parse-and-sum against CORE.** Parse-and-sum recovers the members of each presented option from its text where the parser succeeds (Sec. I), asks the original question and sums the members' probabilities. With 3 orders, it also replaces the model's answer to the original question by the average over three orders. Passes: forward passes per regrouped question (parse-and-sum asks the original question instead of the regrouped one). Add questions (10.9% of all questions) keep the model's own answer. Values $\times 100$; CORE rows are means over 5 seeds on d3-lite and 3 on d3-flash; paired differences with intervals are in Tab. S18. Part.: partition violation; for parse-and-sum, it is 0 by construction with 3 orders and at most the model's order floor with 1 order (d3-lite values in Tab. S4), so this column is not compared across rows.

		d3-lite				d3-flash			
Method	Passes	Acc. ↑	ECE ↓	Brier ↓	Part.	Acc. ↑	ECE ↓	Brier ↓	Part.
base model	1	60.8	5.6	48.4	14.8	72.6	3.5	35.3	15.0
+ parse-and-sum, 1 order	1	63.2	4.5	46.1	2.5	74.4	2.8	33.2	2.8
+ parse-and-sum, 3 orders	3	63.6	4.2	45.9	0.0	75.5	**1.6**	32.8	0.0
CORE	1	67.1	1.7	42.4	5.8	77.2	2.3	30.2	4.8
+ parse-and-sum, 1 order	1	67.5	**1.6**	**41.8**	1.9	**77.5**	2.4	29.9	1.8
+ parse-and-sum, 3 orders	3	**67.7**	2.1	**41.8**	0.0	77.4	2.7	**29.7**	0.0

Table S18. Paired differences for Tab. S17, $\times 100$, with 95% item-bootstrap intervals (10,000 resamples; seeds averaged first). (1) and (3) denote parse-and-sum over 1 and 3 orders. Positive Acc. and negative ECE and Brier differences favor the first term.

	d3-lite			d3-flash		
Difference	Acc.	ECE	Brier	Acc.	ECE	Brier
CORE − base (3)	+3.6 [+2.5, +4.7]	−2.5 [−3.4, −0.4]	−3.4 [−4.2, −2.7]	+1.7 [+0.9, +2.6]	+0.7 [−1.1, +1.4]	−2.6 [−3.2, −2.0]
CORE − base (1)	+4.0 [+2.9, +5.0]	−2.9 [−3.7, −0.5]	−3.6 [−4.4, −2.9]	+2.8 [+1.9, +3.7]	−0.5 [−1.8, +1.1]	−3.1 [−3.7, −2.4]
CORE (3) − CORE	+0.5 [+0.1, +1.0]	+0.4 [+0.1, +1.2]	−0.6 [−0.9, −0.4]	+0.2 [−0.2, +0.6]	+0.4 [+0.1, +1.2]	−0.4 [−0.6, −0.2]

Results. Parse-and-sum improves the accuracy, ECE and Brier score of the base model (Tab. S17). With 3 orders, accuracy rises from 60.8 to 63.6 and ECE falls from 5.6 to 4.2 on d3-lite; on d3-flash accuracy rises from 72.6 to 75.5 and ECE falls from 3.5 to 1.6. CORE is more accurate than the base model with parse-and-sum over 3 orders, by 3.6 points [+2.5, +4.7] on d3-lite and 1.7 [+0.9, +2.6] on d3-flash, and its Brier score is lower on both models (Tab. S18). Its ECE is lower on d3-lite (-2.5 [-3.4, -0.4]); on d3-flash the interval of the difference (+0.7 [-1.1, +1.4]) includes zero. On d3-lite, CORE is more accurate on every question type, with the largest difference on the original questions (64.6 against 59.6), whose answers parse-and-sum reuses for every regrouped question. On d3-flash, the intervals of the accuracy differences on pair merges and splits include zero. Applied to CORE, parse-and-sum over 3 orders raises accuracy by 0.5 points [+0.1, +1.0] on d3-lite and lowers the Brier score on both models, but raises ECE by 0.4 points [+0.1, +1.2] on d3-lite and 0.4 [+0.1, +1.2] on d3-flash.

With correctly parsed members, parse-and-sum over 3 orders has a partition violation of 0 by construction. Over 1 order, its partition violation on d3-lite (2.5 for the base model and 1.9 for CORE) is at most the order floor of Tab. S4, because the TV distance between two distributions cannot grow when both are summed over the same blocks. These partition violations describe the procedure rather than the model, so Tab. S17 does not compare them across rows.

J. Sampled reward on large answer sets

The reward of main-paper Eq. (3) needs only the probabilities of the answers drawn. We test it on a model whose answer set contains every token string: Qwen3-VL-2B-Instruct answers by writing the text of an option. The prompt contains the images, the question, one option text per line and the line "Answer with the text of one option from the list above, copied exactly." For a presented option B, let s_B be its text followed by the end-of-turn token. The model's probability $p(s)$ of a string s is the product of its token probabilities, and the frozen model's is $p_0(s)$. P_0 is the frozen model's probability of the original option strings, renormalized over the options and averaged over the two training orders, and q_f is the target of main-paper Eq. (2) with $w = 0.3$ and $\epsilon = 0.05$. Over strings, the target is

$$q(s) = 0.95\, q_f(B)\, [s = s_B] + 0.05\, p_0(s),$$

which sums to one over all strings. CORE draws $K = 16$ strings of at most 48 tokens per question from the model at temperature 1 and uses the reward $\log q(s) - \log p(s)$, both terms computed by teacher forcing. A string truncated at 48 tokens is rewarded as the event that the answer starts with it. The KL divergence to q cannot be computed, be-

Table S19. **Qwen3-VL-2B-Instruct answering with the option text.** All metrics use the probabilities of the option strings, renormalized over the presented options; Tab. S20 gives the share of the model's probability on these strings. Values $\times 100$; mean $\pm$ 95% Student-t interval over 3 seeds; the base model is evaluated once. Hard: the 25% of items with the highest mean violation over the five kinds of regrouping (Sec. B) under the base model.

	Acc.↑	Orig. acc.↑	ECE↓	Brier↓	Partition↓		Order↓	IIA↓	
					all	hard		add	drop
Base model	50.2	51.6	24.2	71.2	37.3	58.3	21.7	20.3	20.5
Cross-entropy on option strings	59.2 $\pm$ 1.3	55.5 $\pm$ 2.3	4.4 $\pm$ 2.8	50.6 $\pm$ 0.9	11.4 $\pm$ 0.8	15.2 $\pm$ 1.0	9.5 $\pm$ 0.9	10.4 $\pm$ 0.5	10.4 $\pm$ 1.1
CORE, sampled reward	58.0 $\pm$ 1.0	54.5 $\pm$ 1.8	10.0 $\pm$ 4.0	54.0 $\pm$ 1.6	13.6 $\pm$ 1.4	20.1 $\pm$ 2.2	12.4 $\pm$ 1.8	13.0 $\pm$ 1.3	12.9 $\pm$ 1.9

Table S20. Share of the model's probability on the presented option strings, by question type ($\times 100$; mean over questions and seeds). Orig.: original questions in three orders; Pair: pair merges; Split: splits.

	All	Orig.	Pair	Split	Drop	Add
Base model	78.7	93.1	64.0	43.8	89.2	91.1
Cross-entropy	98.8	99.1	98.5	98.3	98.7	98.8
CORE	98.2	98.9	97.5	97.1	98.1	98.7

cause it sums over all strings. The comparison method minimizes the cross-entropy $-\sum_B q_f(B) \log p(s_B)$ over the presented option strings; this loss is computable because q_f is supported on these strings, and it omits the $0.05\,p_0$ part of q. Both methods use 3 seeds and the training items, steps, LoRA settings and learning rate of Sec. C. For evaluation, the probabilities $p(s_B)$ of the presented options are renormalized over these options, and the metrics of the main paper are computed from these renormalized probabilities. For the base model, strings sampled from the model equal an option string as often as the summed probability of the option strings predicts: on the 825 questions of every tenth evaluation item, 78.2% of 13,200 samples are option strings, against a summed probability of 78.2%.

Results. The base model assigns 78.7% of its probability to the presented option strings, but only 64.0% on pair merges and 43.8% on splits (Tab. S20). In the samples we inspected, it writes the text of a single member instead, such as "8" for the option "5 or 8". Both trained models assign 97 to 99% to the option strings for every question type (means over seeds), so after training the metrics cover almost all of the model's probability, and part of the drop in partition violation from the base model comes from this shift. With the sampled reward, CORE lowers the partition violation from 37.3 to 13.6, its part above the order floor from 24.3 to 6.2, and ECE from 24.2 to 10.0, and raises accuracy from 50.2 to 58.0 (Tab. S19). The cross-entropy on the option strings is better on every metric: CORE's partition violation is higher by +2.2 points [1.9, 2.5], its ECE by +5.6 [4.7, 6.1], and its accuracy difference is -1.2 [-1.8, -0.6]. The sampled reward thus trained the model over all token strings, a set too large to sum over, using only the drawn strings and the option strings. An exact alternative nevertheless existed in this setting, because the target assigns at least 95% of its mass to the option strings, and this alternative was better on every metric. The two methods also differ in their targets, since the cross-entropy omits the $0.05\,p_0$ part of q. We did not test a target whose support cannot be enumerated. These results agree with main-paper Sec. 5.4: the target, not the sampled estimator, reduces the violation, and where an exact objective is available it works at least as well.

K. CORE on generative VLMs

We train two generative VLMs, Qwen3-VL-2B-Instruct and Qwen3-VL-8B-Instruct, with CORE, SFT and SFT†. The models answer as in Tab. S6: the prompt labels the options with the letters A, B, ... and asks for one letter, and the answer distribution is the softmax over the logits of the option letters at the first answer token. Training items, steps, LoRA settings, optimizer, targets and losses are those of Secs. C and D; LoRA is applied to all linear layers of the language model, and only the LoRA adapters are trained. All three methods use 3 seeds and the d3-lite learning rate $3 \cdot 10^{-5}$, without tuning on the VLMs. The 2B models were trained and evaluated on RTX 4090 GPUs and the 8B models on H100 GPUs. We re-evaluated both base models with the code and GPUs of the trained models, so their rows in Tab. S21 differ from those in Tab. S6 by at most 0.3 points.

Results. Table S21 gives all metrics and Tab. S22 the paired differences from CORE. On both models, CORE has the lowest partition violation of all rows and is less accurate than SFT and SFT†. CORE's partition violation is 9.6 against 13.2 for SFT† on the 2B model and 10.3 against 11.4 on the 8B model. On the 2B model, CORE also has the lowest ECE, order TV and IIA TV, and the intervals of its differences from SFT† in partition violation, ECE and order TV exclude zero. On the 8B model, the differences between CORE and SFT† are smaller. The partition violation of SFT† is higher by +1.1 points [0.7, 1.5], and SFT† has the lower Brier score (-3.2 [-4.8, -1.6]). The intervals of the differences in ECE (+1.1 [-0.1, 2.4]), in partition violation on

Table S21. **CORE on generative VLMs scored by the probability of the option letter.** All methods train LoRA adapters of Qwen3-VL-Instruct with the settings of Secs. C and D and learning rate $3 \cdot 10^{-5}$. Values $\times 100$; mean $\pm$ 95% Student-t interval over 3 seeds; the base model is evaluated once. Hard: the 25% of items with the highest mean violation over the five kinds of regrouping (Sec. B) under each base model. Orig. acc.: accuracy of the order-averaged answer to the original question.

	Acc.↑	Orig. acc.↑	ECE↓	Brier↓	Partition↓		Order↓	IIA↓	
					all	hard		add	drop
Qwen3-VL-2B-Instruct									
Base model	53.2	48.4	18.8	62.2	21.7	41.5	24.5	21.5	20.1
SFT	61.0 ± 0.8	59.2 ± 1.3	18.5 ± 7.2	55.3 ± 5.1	19.7 ± 3.5	27.3 ± 4.4	15.8 ± 3.4	15.4 ± 3.8	14.8 ± 3.0
SFT†	62.5 ± 0.5	59.4 ± 1.0	14.6 ± 11.2	51.4 ± 7.0	13.2 ± 3.1	21.6 ± 5.6	13.4 ± 3.4	12.3 ± 2.6	12.8 ± 3.1
CORE	58.8 ± 0.6	54.4 ± 1.0	6.7 ± 2.6	51.4 ± 0.4	9.6 ± 0.7	15.5 ± 2.1	11.6 ± 1.4	10.6 ± 0.6	9.2 ± 0.6
Qwen3-VL-8B-Instruct									
Base model	62.2	60.4	23.3	57.7	23.5	46.1	22.6	21.9	20.9
SFT	69.0 ± 0.8	68.4 ± 2.6	15.9 ± 4.0	45.5 ± 2.8	17.8 ± 0.8	28.9 ± 3.2	13.9 ± 3.0	13.1 ± 3.6	14.4 ± 3.5
SFT†	70.8 ± 0.8	69.6 ± 2.0	10.7 ± 0.9	40.6 ± 1.2	11.4 ± 0.6	19.5 ± 1.2	12.0 ± 1.1	11.1 ± 1.0	12.6 ± 1.2
CORE	67.2 ± 0.5	63.2 ± 1.2	9.6 ± 3.7	43.8 ± 1.2	10.3 ± 1.9	18.8 ± 4.5	11.5 ± 1.6	10.5 ± 1.4	11.0 ± 1.8

Table S22. **Paired differences from CORE on generative VLMs.** Values are each method minus CORE, $\times 100$, with paired item-bootstrap 95% intervals (10,000 resamples; seeds averaged per method first). Positive values mean higher accuracy, or higher ECE, Brier score, partition violation or order TV, than CORE. Above floor: partition violation minus the order floor of Tab. S4.

	Acc.	Orig. acc.	ECE	Brier	Partition		Order	Above floor
					all	hard		
Qwen3-VL-2B-Instruct								
Base model	−5.6 [−6.7, −4.5]	−6.0 [−8.1, −3.9]	+12.0 [10.7, 13.3]	+10.8 [9.7, 11.9]	+12.2 [11.4, 13.0]	+26.0 [24.4, 27.7]	+12.8 [12.0, 13.7]	+4.2 [3.6, 4.9]
SFT	+2.1 [0.8, 3.5]	+4.8 [2.7, 7.0]	+11.8 [10.3, 13.1]	+3.9 [2.1, 5.7]	+10.1 [9.4, 10.9]	+11.8 [10.5, 13.2]	+4.2 [3.6, 4.8]	+7.6 [6.9, 8.3]
SFT†	+3.6 [2.3, 5.0]	+5.0 [3.0, 6.9]	+7.9 [6.5, 9.1]	+0.1 [−1.7, 1.8]	+3.6 [3.2, 4.1]	+6.1 [4.9, 7.3]	+1.8 [1.2, 2.3]	+2.6 [2.2, 3.0]
Qwen3-VL-8B-Instruct								
Base model	−5.0 [−5.9, −4.1]	−2.9 [−4.5, −1.2]	+13.7 [12.6, 14.7]	+13.9 [12.6, 15.2]	+13.2 [12.3, 14.1]	+27.2 [25.7, 28.8]	+11.1 [10.1, 12.1]	+6.2 [5.4, 7.1]
SFT	+1.8 [0.6, 3.1]	+5.2 [3.3, 7.0]	+6.3 [5.0, 7.5]	+1.7 [0.1, 3.3]	+7.5 [6.9, 8.2]	+10.0 [8.6, 11.4]	+2.4 [1.8, 3.0]	+6.0 [5.4, 6.7]
SFT†	+3.7 [2.4, 4.9]	+6.3 [4.4, 8.3]	+1.1 [−0.1, 2.4]	−3.2 [−4.8, −1.6]	+1.1 [0.7, 1.5]	+0.7 [−0.3, 1.7]	+0.5 [−0.1, 1.0]	+0.8 [0.4, 1.2]

the hard items and in order TV include zero.

CORE's accuracy (58.8 and 67.2) is above that of the base models (53.2 and 62.2). SFT† exceeds CORE by +3.6 points [2.3, 5.0] on the 2B model and +3.7 [2.4, 4.9] on the 8B model, and SFT exceeds it by +2.1 [0.8, 3.5] and +1.8 [0.6, 3.1]. On d3-lite, SFT† also exceeds CORE in accuracy, but SFT does not (Tab. S8).

Order sensitivity. The base VLMs are sensitive to the option order (order TV 24.5 and 22.6), so a large part of their partition violation is the order floor of Tab. S4. A model that satisfies main-paper Eq. (1) exactly but has the same order sensitivity also has this floor. The part above the floor is 6.9 of 21.7 points for the 2B base model and 9.7 of 23.5 for the 8B base model. CORE lowers both parts: its order floor is 6.9 and 6.8 points, and its partition violation above the floor is 2.6 and 3.4, the lowest of all rows (last column of Tab. S22). SFT lowers the order floor to 9.5 and 8.3 points, but its partition violation above the floor, 10.2 and 9.5 points, is close to or above that of the base models. SFT's mean additivity error, −7.5 and −8.4 points against −1.1 and −5.7 for the base models, shows that SFT makes both models more under-additive.

L. Qualitative examples and failure cases

Figure S2 shows four evaluation items answered by d3-lite before training (Base) and after training with CORE. Each row shows the original question and one pair merge of it. The four pair merges are selected from the 2,100 pair merges of the evaluation set by fixed rules: (a) the largest reduction of the partition violation among pair merges on which the base model is under-additive and CORE is coherent (mean partition violation over seeds at most 0.05); (b) the most negative additivity error under CORE; (c) the most positive additivity error under CORE; (d) the highest probability of a wrong option among pair merges on which CORE is coherent and its most probable original option is wrong in all seeds. Each rule takes the highest-ranked pair merge from a benchmark not used by an earlier rule, and the item of main-paper Fig. 1 is excluded. Rankings use the mean over the five seeds of CORE; the figure shows the seed whose partition violation on the shown question is the median of the five.

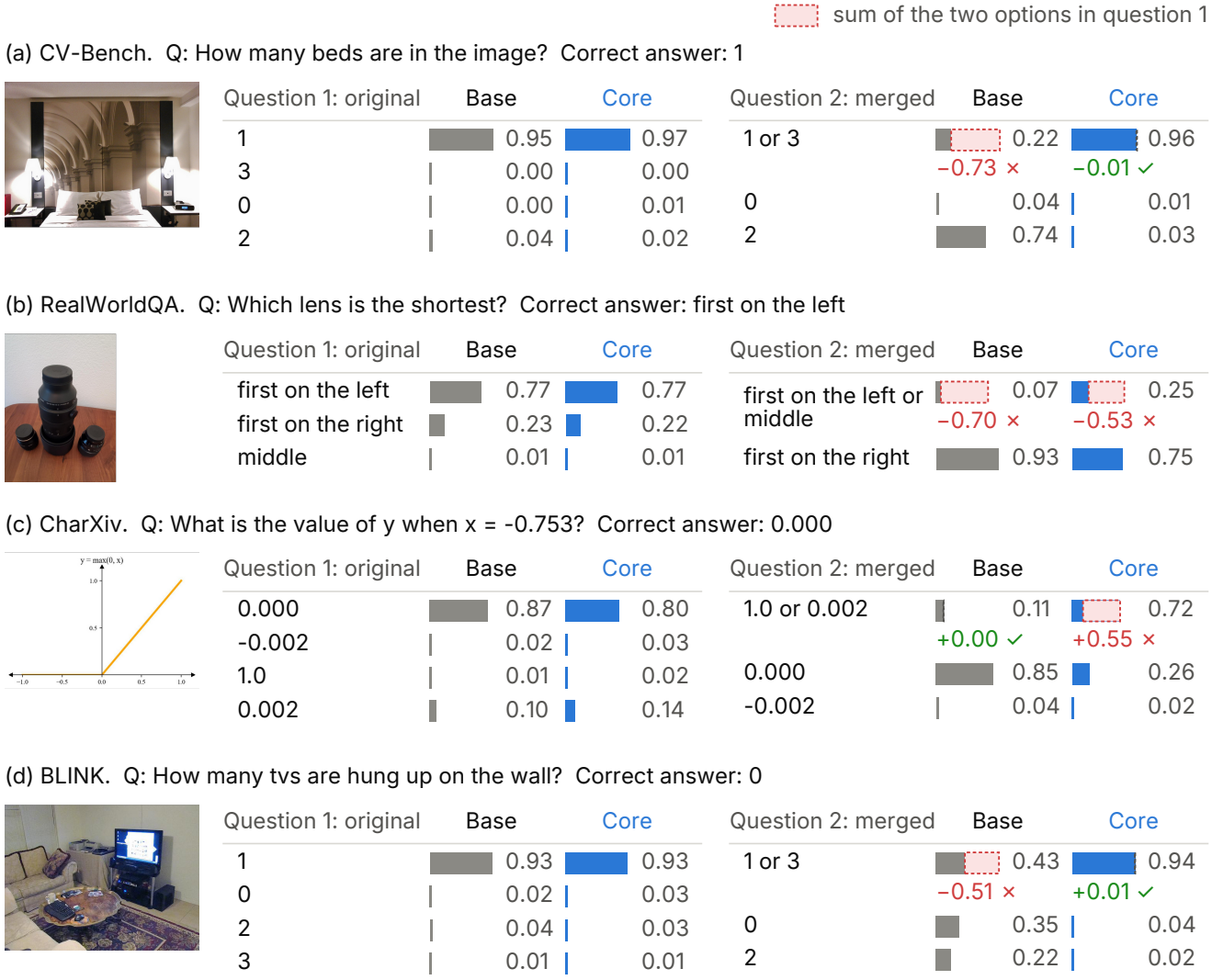


Figure S2. **One success and three failure cases of CORE on d3-lite, selected by fixed rules.** Each row shows an evaluation item, its original question (question 1, averaged over three option orders) and one pair merge (question 2), answered by the base model and by CORE (the seed whose partition violation on question 2 is the median of five seeds). (a) Largest reduction of the partition violation among pair merges on which the base model is under-additive and CORE is coherent. (b) Most negative additivity error under CORE. (c) Most positive additivity error under CORE. (d) Highest probability of a wrong option among pair merges on which CORE is coherent and wrong on question 1 in all seeds. Section L gives the rules. Dashed boxes span from the merged option's probability to the sum of its members' probabilities in question 1; the number below each box is the additivity error, marked × when its absolute value exceeds 0.1 and ✓ otherwise. Option texts in (b) are abbreviated.

Success. In (a), the base model assigns 0.22 to the merged option "1 or 3", although its members receive 0.95 in the original question. CORE assigns it 0.96, with an absolute additivity error below 0.03 in every seed.

Remaining under-additivity. In (b), the merged option contains the correct answer, and its members receive 0.75 to 0.78 in the original question under both models. The base model assigns it 0.07 and CORE 0.25 (CORE's additivity error is -0.48 to -0.54 over seeds). Both models answer the original question correctly but choose the unmerged wrong option in the regrouped question.

Over-additivity. In (c), the base model is coherent (additivity error $+0.00$). CORE assigns 0.72 to the merged option "1.0 or 0.002", although its members receive 0.16 in the original question (additivity error $+0.40$ to $+0.57$ over seeds). CORE answers the original question correctly with probability 0.80 but assigns 0.26 to the correct option in the regrouped question. Over all 2,100 pair merges, the base

Table S23. **ECE binning and partition violation by regrouping type.** Each row gives CORE, the base model, the trained baseline (published or † setting) with the lowest value, and CORE minus that baseline with its 95% paired item-bootstrap interval (10,000 resamples); training-free corrections are discussed in Sec. M. Values are ×100 (lower is better) and means over 3 seeds per baseline and over 5 (d3-lite) or 3 (d3-flash) for CORE. Adaptive ECE uses 15 bins with equal numbers of questions. There are 2,100 pair-merge and 1,050 split questions. *: the variant used in the main paper.

	d3-lite				d3-flash			
Metric	CORE	Base	Best trained baseline	CORE − it	CORE	Base	Best trained baseline	CORE − it
ECE, 10 bins	1.6	5.6	LLM-as-Jev (1.7)	−0.2 [−0.9, +1.0]	2.2	3.5	LLM-as-Jev (1.8)	+0.5 [−0.4, +1.4]
ECE, 15 bins*	1.7	5.6	LLM-as-Jev (1.9)	−0.2 [−1.0, +0.9]	2.3	3.5	LLM-as-Jev (1.8)	+0.5 [−0.3, +1.4]
ECE, 20 bins	2.1	5.6	LLM-as-Jev (2.3)	−0.2 [−1.0, +0.8]	2.6	3.6	LLM-as-Jev (2.2)	+0.4 [−0.4, +1.2]
Adaptive ECE, 15 bins	1.8	5.6	LLM-as-Jev (2.2)	−0.4 [−1.0, +0.8]	2.3	3.8	LLM-as-Jev (1.8)	+0.4 [−0.4, +1.4]
Partition, pooled*	5.8	14.8	LLM-as-Jev† (9.4)	−3.5 [−3.8, −3.3]	4.8	15.0	SFT† (7.3)	−2.4 [−3.0, −1.9]
Partition, pair merges	5.7	14.0	LLM-as-Jev† (9.3)	−3.5 [−3.8, −3.3]	4.7	15.4	SFT† (7.2)	−2.5 [−3.1, −1.9]
Partition, splits	6.1	16.4	LLM-as-Jev† (9.6)	−3.5 [−3.9, −3.1]	5.1	14.2	SFT† (7.4)	−2.4 [−3.1, −1.6]

Table S24. Training and inference cost. Prompts per training item: questions passed through the frozen model without gradient (frozen) and forward and backward through the LoRA adapters (trained), with 4 items per step. The K answers of the RL methods are sampled from the computed distribution and need no further pass. Training time: median over seeds of one 600-step run without the intermediate evaluations, divided by the time of a run with the passes of CORE on the same GPU type under the same load. CORE takes 29 min on d3-lite (one RTX 4090) and 56 min on d3-flash (one H100). Inference: forward passes per question.

	Prompts per item	Training time ratio		Passes per
Method	frozen / trained	d3-lite	d3-flash	question
CORE	2 / 4	1.00	1.00	1
SFT	0 / 2	0.45	0.57	1
SFT†	0 / 4	0.72	0.84	1
LLM-as-Jev	2 / 2	0.63	0.74	1
LLM-as-Jev†	2 / 4	1.11	1.00	1
ACRE	2 / 4[a]	0.99	1.03	1
ACRE†	2 / 4	1.05	–	1
PA-GRPO	4.9 / 4.9	1.45[b]	1.30	1
PA-GRPO†	4.9 / 6.9	1.72[b]	–	1
exact $\mathrm{KL}(p_f \Vert q_f)$	2 / 4	1.00	1.01	1
cross-entropy to q_f	2 / 4	0.99	–	1
Temperature scaling	–	–[e]	–[e]	1
Sorted options	–	–	–	1
Order averaging	–	–	–	2
OBS	–	–	–	c (3.75)[c]
Parse-and-sum, 1 order	–	–	–	1[d]
Parse-and-sum, 3 orders	–	–	–	3[d]

[a]The ACRE loss uses only the two original orders; our implementation also passes the two regrouped questions of each item, which a standalone implementation of ACRE would skip (2 / 2 prompts per item). [b]PA-GRPO ran with gradient checkpointing (one extra forward pass in the backward pass), but its d3-lite reference runs on the same GPU did not; on d3-flash, all runs used gradient checkpointing. [c]OBS asks one yes/no question per presented option; c is the number of presented options, with its mean over the 8,250 evaluation questions in parentheses. [d]For each regrouped question, the original question is asked in 1 or 3 orders instead; add questions keep the model's own answer (1 pass). [e]The temperature is fitted with one pass over 400 training items.

model and CORE have mean absolute additivity errors of 0.13 and 0.05, and 1,013 and 305 merges with an error beyond ±0.1 (means over seeds). The base model's large errors are almost all negative (984 below −0.1, 29 above 0.1); CORE's are split between 160 below −0.1 and 145 above 0.1 (this count ranges from 78 to 198 over seeds), so CORE over-corrects some merged options.

Coherent but wrong. In (d), the television stands on a cabinet, and the correct answer is 0. Both models assign

0.93 to "1" in the original question. The base model assigns 0.43 to "1 or 3" and CORE 0.94, while its members receive 0.94 under both models. Coherence constrains how a model's probabilities relate across regroupings of the options; it does not make the answer to the original question correct.

M. Robustness of the metrics

Two metric choices could affect the conclusions of the main paper: ECE uses 15 bins of equal width, and the partition violation pools pair merges and splits. Table S23 varies both. With 10 or 20 bins of equal width, or with 15 bins that hold equal numbers of questions (adaptive ECE, the equal-count binning of [24] applied to the most probable option), no trained baseline changes its rank relative to CORE. With these binnings, as with 15 bins, the difference between CORE and the trained baseline with the lowest ECE has an interval that includes zero. For the training-free corrections of main-paper Table 2, the comparison depends on the binning: temperature scaling on d3-lite and order averaging on d3-flash have a lower ECE than CORE with some binnings and a higher one with others, and all eight of these differences have 95% intervals that include zero. Fitted on the evaluation questions themselves, which favors it, temperature scaling has ECE 1.9 (d3-lite) and 2.4 (d3-flash) and partition violation 14.6 on both models; fitted on the training items as in main-paper Table 2, it has ECE 1.9 and 2.7 and partition violation 14.6 and 14.3.

On both models, CORE has the lowest partition violation of all methods in Tab. S23 on pair merges and on splits, and for each regrouping type its difference from the best trained baseline has an interval that excludes zero.

N. Computational cost

CORE adds no inference cost: like every trained baseline, it answers a question with one forward pass. Order averaging needs 2 passes, OBS needs one per presented option (3.75 on average on our evaluation set), and parse-and-sum needs a parser, access to the original question and 1 pass per regrouped question with 1 order or 3 passes with 3 orders. On d3-flash (one H100), evaluating the 8,250 questions, including model loading, takes 12 min with one pass per question (measured with sorted options), 22 min with order averaging and 73 min with OBS. In training, CORE passes the two training orders of each item through the frozen model without gradient and all four questions of the item through the adapters. A d3-flash run of CORE trains for 56 min on one H100 without the intermediate evaluations (68–69 min with them, Sec. C); LLM-as-Jev† and ACRE take about as long (time ratios 1.00 and 1.03). SFT and LLM-as-Jev, trained on the original questions only, take 0.57 and 0.74 times as long, and PA-GRPO, which passes 4.9 option orders per item on average, takes 1.30 times as long. Training and inference costs of all methods are in Tab. S24.