SAMPLE FREQUENCIES ARE NOT PROBABILITIES: LABEL-FREE THRESHOLDS FOR SET-VALUED SELF-CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Drawing several samples and combining them is standard test-time practice. When outputs are sets (entity mentions, relation triples, labels, keyphrases), combining means keeping the candidates that appear in more than a fraction θ of samples, and practice defaults to $\theta = 1/2$. Across 246 model–dataset pairs, 31 checkpoints and five kinds of set-valued output, the F_1-optimal threshold lies below one half in 192. The reason is that a sample frequency is not a probability of correctness: every task family we measure is overconfident, and a candidate emitted in nearly every sample is right far less than always. We show that *sampling sharpness* explains this miscalibration across every intervention we can make. Temperature and instruction tuning are two routes to it, not two mechanisms: matching tuned and base checkpoints on sharpness removes the apparent tuning gap (from +0.104 to +0.007, interval spanning zero). Because sharpness is visible in unlabeled samples, the threshold can be read off samples already drawn. A three-feature sharpness predictor, with zero labels, beats a grid search on ten labeled examples (+0.013 F_1, interval above zero), stays ahead when refit on held-out task families, and never falls below majority voting on a newer model generation it never saw. The same statistics say when aggregation is worth running at all: its gain concentrates on weak extractors (+0.072 F_1 on high-disagreement cells), is negligible on strong ones, and what it buys is precision, not recall.

1 INTRODUCTION

Drawing k samples and combining them is the usual way to spend extra compute at test time (Wang et al., 2023; Snell et al., 2024; Brown et al., 2024). When a task has a single answer, combining means taking a vote, and the analysis is old: if each sample is right with probability p, the vote is right with probability $\Pr[\mathrm{Binom}(k, p) > k/2]$ (de Condorcet, 1785; Kuncheva, 2004).

Many tasks do not have a single answer. Ask a model for the named entities in a sentence and it returns a set. Combining samples now means something different: each candidate m appears in some fraction f_m of the k samples, and you keep it if f_m is large enough. This is what people do; Xie et al. (2023) keep a mention that "appeared in more than half of the responses." The same rule applies to any output that is a set: extracted entities and relations, multi-label moderation tags, keyphrases, the fields of a structured record, or a list of retrieved items. But "large enough" is a number, and one half came from the single-answer case, where it was never a choice. Once the output is a set, the knob exists and somebody has to set it. The knob is specific to set-valued outputs. Calibrating a whole answer from sample agreement (Lyu et al., 2025) has no inclusion threshold, and atomic self-consistency (Thirukovalluru et al., 2024) applies a support threshold to units of long-form text but treats it as a setting to tune. We ask where the threshold belongs, why the default is wrong, and how to set it without labels.

Where the threshold belongs. Suppose at threshold θ you emit P items, TP of them correct, out of G correct items in total, so $F_1 = 2TP/(P + G)$. Loosen the threshold slightly so dN more candidates are emitted, a fraction q of them correct. Then

$$dF_1 = \frac{2\,dN\,[\,q(P+G) - TP\,]}{(P+G)^2},$$

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

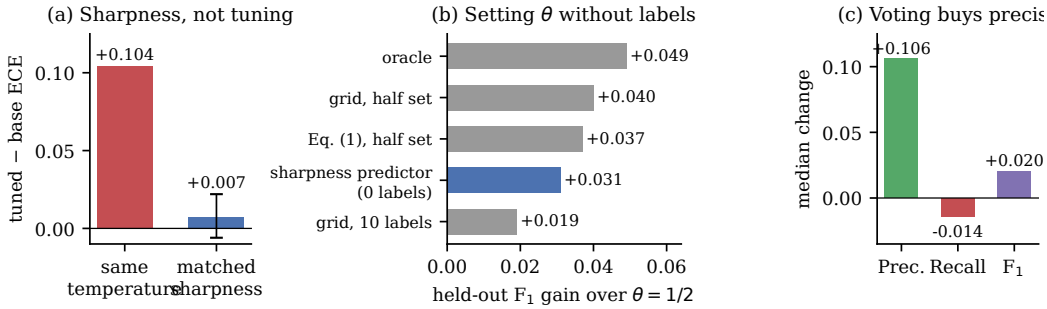


Figure 1: **The paper at a glance.** (a) Instruction-tuned models look worse calibrated than their base checkpoints at the same temperature, but the gap vanishes once the two are matched on sampling sharpness (§4). (b) A threshold predicted from unlabeled samples beats a grid search that uses ten labels (§5). (c) Across 174 cells, aggregation raises precision and leaves recall flat (§6).

which is zero exactly when

$$q(\theta^*) = \mathrm{F}_1^*/2. \tag{1}$$

Keep loosening the threshold as long as the candidates you let in are right at least half as often as your current score. A system at $\mathrm{F}_1 = 0.6$ should accept anything with better than a 0.3 chance of being right; demanding a majority is appropriate only for a perfect system. On a calibrated score this is the standard F-measure thresholding result (Zhao et al., 2013; Lipton et al., 2014), and we take it as given.

What does not carry over. Those results threshold a calibrated probability. A sample frequency is not one. So the plug-in rule they license (cut the score at $\mathrm{F}_1^*/2$) fails on raw frequencies, and it fails worst where miscalibration is largest. This paper measures that miscalibration across set-valued tasks, identifies what explains it, and turns that into a threshold you can set without labels.

Contributions.

1. **A measurement.** Over 246 model–dataset pairs, sample frequencies are systematically overconfident, and the F_1-optimal threshold lies below one half in 192 (§3).
2. **A mechanism.** Sampling sharpness explains the miscalibration across our interventions. Temperature and instruction tuning are two routes to it; matched on sharpness, the tuning effect vanishes (§4).
3. **A label-free procedure.** A sharpness predictor sets θ from unlabeled samples, beats a ten-label grid search, and transfers across task families and to a newer model generation; a disagreement statistic says whether to aggregate at all (§5).
4. **What aggregation buys.** Thresholding buys precision, not recall: precision rises in 170 of 174 cells while recall is flat (§6).

2 Experimental Setup

Models. 31 checkpoints from 7 families: Qwen2.5 (Yang et al., 2024) at 0.5B–14B, OLMo-2 at 7B and 13B (Team OLMo, 2025), Mistral-7B (Jiang et al., 2023), Falcon3 at 3B and 7B, Granite-3.1 at 2B and 8B, SmolLM2 at 360M and 1.7B, each in base and instruction-tuned form, plus Phi-3.5-mini-instruct (Abdin et al., 2024). The base/instruct pairs separate instruction tuning from family and scale. Qwen3-8B and Qwen3-32B are held out entirely for the transfer test of §5.

Tasks. Nine evaluation sets over five kinds of set-valued output: flat NER (CoNLL-2003, Tjong Kim Sang & De Meulder, 2003; WikiANN in German, Spanish, Chinese and Arabic, Pan et al., 2017), fine-grained NER (Few-NERD, 66 types; Ding et al., 2021), relation triples (CoNLL04; Roth & Yih, 2004), multi-label classification (GoEmotions; Demszky et al., 2020) and keyphrases (Inspec; Hulth, 2003). Sources, splits, prompts and metrics are in Appendix B.3.

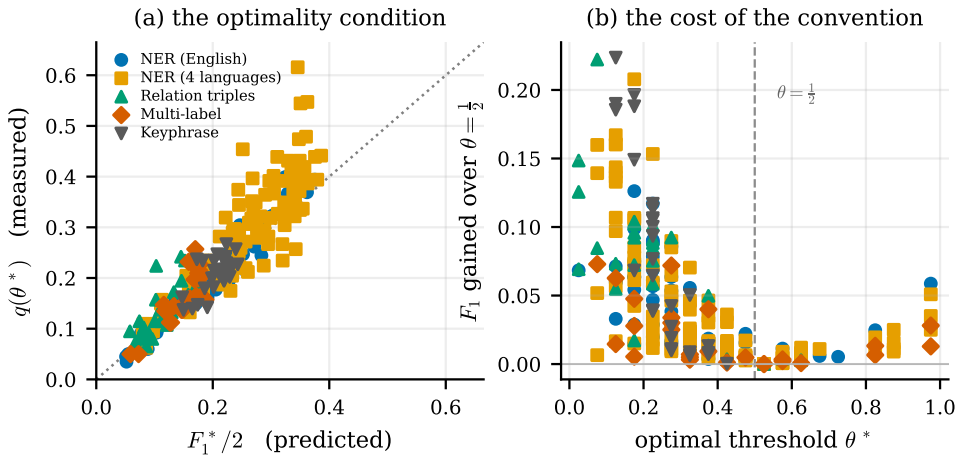


Figure 2: (a) Measured precision at the optimal cut against $F_1^*/2$, one point per cell, five task families; dotted line $y = x$, nothing fitted. Agreement is partly mechanical (Appendix D.8). (b) F_1 lost by using $\theta = 1/2$ grows as the optimum moves below one half.

Protocol. Every cell generates one greedy continuation and 20 samples at $\tau = 0.7$; every $k \in \{1, 2, 3, 5, 10, 20\}$ is a prefix of that single list, so k varies alone. Few-shot exemplars come from the training split and are filtered so no exemplar's text appears in the evaluation set. The resampling unit is the example; intervals are 2,000-replicate percentile bootstraps, paired across k and across methods.

Sharpness and disagreement. We summarize a cell's sampling distribution by statistics computed without labels: the mean and median candidate frequency and the *mass at* $f = 1$, the share of candidates emitted in every sample. Higher values mean a sharper, more concentrated sampler. *Disagreement* is the share of candidate mass appearing in at most half the samples.

3 The Optimal Threshold Is Usually Below One Half

Fix a model, dataset and k. Let S be the candidates produced in at least one sample. A threshold rule keeps $\{m : f_m > \theta\}$. Majority voting is $\theta = 1/2$, the union of all samples is $\theta \to 0$, and unanimity is $\theta \to 1$: three points on one curve. Raising θ only removes candidates, so recall falls as θ rises and can never exceed the share of gold items in S.

For each cell we sweep θ over achievable values to find θ^* and F_1^*, and report the 246 cells with an interior optimum (exclusions in Appendix D.6). The optimum lies below one half in 192 of them. The measured precision q of candidates near the cut tracks $F_1^*/2$ closely (correlation 0.91, MAE 0.03; Figure 2a). We do not treat this agreement as evidence: the cut is an argmax located on the same curve, and a mechanical null in which only the ranking is informative fits at least as closely (Appendix D.8). The evidence is held-out: a threshold chosen on one half of the examples beats one half on the other half in 189 of 246 cells (§5). Cells whose optimum lies furthest below one half lose the most by using it (Figure 2b).

Note that Equation 1 constrains the precision at the cut, not θ itself: substituting one for the other (the naive plug-in $\theta^* = F_1^*/2$) is $5.0\times$ less accurate. The measurement does not hinge on how wide a band of candidates we average over: MAE stays between 0.027 and 0.046 for bands of 50 to 2,000 candidates (Appendix D.9). Some appendix analyses use a slightly different set of 259 scored cells; the exclusion rules are in Appendices D.6 and L.

Table 1: **Sharpness, not tuning.** Tuned-minus-base calibration error. At a common nominal temperature tuning looks harmful; once each tuned cell is matched to its own base checkpoint at equal sharpness, the gap disappears. Interval is checkpoint-clustered.

Comparison	Tuned − base ECE	Tuned worse in
Matched family/size/dataset, τ=0.7	+0.091	113/124
Checkpoint level, τ=0.7	–	14/14 (p=0.0001)
Common nominal τ (sweep)	+0.104	–
Matched on sharpness	+0.007 $[-0.006, +0.022]$	–
Mass at $f = 1$ (base → tuned)	4.1% → 16.3%	

4 Sampling Sharpness Explains Where the Threshold Belongs

Frequencies are overconfident. Binning candidates by frequency and asking how often each bin is right, calibration error runs 0.056–0.169 across the five task families. The miscalibration is worst exactly where practice cuts: candidates emitted in nearly every sample are right far less than always (top-bin gap -0.279 to -0.597; Appendix D.3). That sample frequencies are miscalibrated is not itself new (Kuleshov & Liang, 2015; Jagannatha & Yu, 2020; Lyu et al., 2025; Huang et al., 2024); what we add is what explains it, and therefore where the set-valued threshold belongs.

Instruction tuning looks like the cause. Instruction-tuned checkpoints are $2.4\times$ worse calibrated. On 124 pairs matched on family, size and dataset the tuned model is worse in 113 (mean ΔECE +0.091); at the checkpoint level it is worse in 14 of 14 (exact sign-flip $p = 0.0001$). Controlling for F_1 leaves the gap intact, so sharpness is not a stand-in for how accurate a model is. The clue is where the mass goes: tuning raises the share of candidates emitted in every sample from 4.1% to 16.3%. All of them carry $f = 1$, so no threshold can separate them, and they are wrong often enough to matter.

But the cause is sharpness. Concentrating the sampling distribution is exactly what lowering the temperature does. Sweeping $\tau \in \{0.3, 0.5, 0.7, 1.0, 1.3\}$ for 14 base/instruct pairs across 6 families, temperature moves calibration error *within* a checkpoint more than tuning moves it *between* checkpoints, and drags θ^* with it in the same direction in 14 of 14 pairs. We then match each tuned cell to its own base checkpoint at the temperature that equalizes sharpness. The tuning gap collapses from +0.104 at a common nominal τ to +0.007, with a checkpoint-clustered interval of $[-0.006, +0.022]$ (Table 1).

The unmatched cells. 114 tuned cells are sharper than their base checkpoint at any temperature in $[0.3, 1.3]$, across all four task families, so they cannot enter the matched comparison. We therefore state the result for what it covers: wherever tuned and base samplers can be made equally sharp, tuning adds no measurable miscalibration. Beyond that range, tuning reaches a sharpness that temperature alone cannot, and there the two cannot be separated with our data. This is the one place where sharpness is a floor on what tuning does rather than a full substitute for it. It does not weaken the practical consequence, because the predictor of §5 reads sharpness directly and does not need to know whether it came from temperature or tuning.

An out-of-sample prediction. The mechanism predicts that a very sharp model should place its threshold *above* one half. Qwen3-8B and Qwen3-32B, the sharpest models we measure (mass at $f = 1$ of 0.46), test that prediction. Their optimal thresholds lie above one half on 4 of 6 cells, and the few multi-label cells whose optimum sits high are the ones with the most mass at $f = 1$. "$\theta^* < 1/2$" is a property of diffuse samplers, not a law. Recalibrating frequencies with isotonic regression on a development half and then applying the textbook plug-in rule closes essentially all of the gap to a held-out grid search, which confirms that miscalibration is what breaks the plug-in rule.

5 Setting the Threshold Without Labels

Because sharpness is visible in the unlabeled samples, a practitioner with no annotation can read the threshold off the samples already drawn.

A sharpness predictor. A linear predictor of θ^* from three sharpness statistics (mean frequency, median frequency, mass at $f = 1$), fit leave-one-cell-out, sets a threshold for each cell with zero labels and gains +0.031 F_1 of the +0.049 an oracle could reach (Table 2). The simplest label-free rule, a constant $\theta = 0.25$, comes within 0.003 F_1 of it on average but, unlike the predictor, falls below majority voting on the sharpest models (Table 3). It beats a grid search on ten labeled examples (+0.013 F_1, interval [+0.011, +0.016]) and ties one on twenty-five. Used as a prior in a shrinkage rule, it is never significantly worse than grid search at any label budget. Refit leave-one-task-family-out, it stays ahead of the ten-label search in all four held-out families, so the sharpness-to-threshold map is a property of the sampling distribution, not of the task. Together with the transfer test below, the predictor is evaluated held out at three levels: by cell, by task family, and by model generation.

Table 2: Held-out F_1 gain over majority voting ($\theta = 1/2$), mean over the 246 cells, by how the threshold is chosen. The predictor-versus-ten-label comparison is paired on a common split.

Threshold chosen by	Labels needed	ΔF_1	Cells improved
Oracle (test-set optimum)	all	+0.049	–
Grid search on half the examples	half the set	+0.040	189/246
Eq. 1 solved on half the examples	half the set	+0.037	188/246
Sharpness predictor	**0**	**+0.031**	**176/246**
Grid search on 10 examples	10	+0.019	–

Whether to aggregate at all. Disagreement tracks extraction quality tightly (correlation -0.72 with F_1), and the gain from choosing θ is concentrated on weak extractors. Splitting cells by the F_1 they reach at $\theta = 1/2$: on the 42 held-out cells at or above 0.4, a ten-label grid search has median gain -0.012 and hurts in 27; on the 40 weaker cells it gains +0.038. Gain and quality correlate at -0.54. A large gain is therefore a diagnostic as much as an improvement: the candidate set is full of things the model produces rarely and gets wrong, which can mean a genuinely hard case or a broken pipeline (Appendix E.5). Figure 3 packages both statistics into a triage rule.

Triage rule: choosing θ from unlabeled samples.

Low disagreement (a working extractor): aggregation buys little (+0.019 F_1; on the strongest cells tuning even hurts). Ship a single sample, or majority vote.

High disagreement (a struggling extractor): first check the pipeline. If the case is genuinely hard, aggregation recovers a large gain (+0.072 F_1). With no labels, set θ with the sharpness predictor; with twenty-five or more labels, grid search.

Figure 3: The label-free triage rule. Two statistics read off samples already drawn (how much they disagree, and how sharp they are) decide whether to aggregate and where to cut.

Transfer to a newer model generation. We freeze the predictor on the 239 older cells ($\leq$14B, $\leq$2024), withhold the whole Qwen3 family, and apply it unchanged (Table 3). It matches or beats majority voting on all six held-out cells, strictly on five, and correctly raises the threshold above one half on the four high-θ^* cells, where the fixed $\theta = 0.25$ shortcut instead falls below majority voting. It misses Few-NERD's unusually low optima. Across the six cells it gains +0.0027 F_1 against +0.0130 for the oracle and -0.0003 for the fixed $\theta = 0.25$: transfer is directional rather than fully calibrated. These sharper models have flat F_1 curves near their optima, so threshold errors are cheap.

6 What Thresholding Buys: Precision, Not Recall

Everything a threshold emits came from the model. Lowering θ keeps more of what was produced and raising it keeps less, but nothing can add a mention the model never produced, so recall has a ceiling no threshold can lift. The direction follows from how thresholds work and has long been known for combining taggers (Màrquez et al., 1998; Florian et al., 2003); what had not been measured is its size. It is the largest and most uniform effect we measure (Appendix K): across 174 cells, aggregation raises precision in 170, significantly in 150 after correction (median +0.106), with only 2 moving significantly the other way, while recall is flat or slightly down (median -0.014). The F_1

Table 3: Frozen-predictor transfer to 2025 models never seen during development. $\hat{\theta}$: predicted threshold. F_1 at $\theta = 0.5$, 0.25, $\hat{\theta}$ and the oracle.

Model	Dataset	θ^*	$\hat{\theta}$	$F_1^{0.5}$	$F_1^{0.25}$	$F_1^{\hat{\theta}}$	oracle
Qwen3-8B	CoNLL	0.78	0.73	0.716	0.708	0.721	0.722
Qwen3-8B	Few-NERD	0.07	0.55	0.313	0.334	0.313	0.343
Qwen3-8B	GoEmotions	0.88	0.58	0.343	0.350	0.347	0.357
Qwen3-32B	CoNLL	0.97	0.72	0.723	0.716	0.727	0.732
Qwen3-32B	Few-NERD	0.17	0.55	0.516	0.523	0.518	0.526
Qwen3-32B	GoEmotions	0.93	0.55	0.331	0.309	0.332	0.340

change is a much smaller +0.020, so reporting only F_1 hides which of the two a deployment is getting, which matters whenever a false positive costs something different from a miss. Two practical corollaries: a deployment with an asymmetric loss wants a different point on the same curve, and when the model is not fixed, a larger model at $k = 1$ usually beats a smaller one at $k = 20$ (7 of 8 pairs; Appendix G).

Measurement practice. Three common practices distort published k-curves for these tasks: scoring format failures as reasoning failures, mixing greedy $k = 1$ with sampled $k > 1$ (worth up to 0.110 F_1), and pooling label spaces of different sizes. Every fit here stays within one (model, dataset) cell and every k comes from one run at one temperature (Appendix I).

7 Related Work

Voting and sampling. The classical theory describes a vote between alternatives, where one half is the definition of majority (de Condorcet, 1785; Lam & Suen, 1997; Kuncheva, 2004; Zhou, 2012). Self-consistency (Wang et al., 2023) and its scaling studies (Snell et al., 2024; Brown et al., 2024) concern a single answer, as do methods that predict or stop sampling (Aggarwal et al., 2023; Li et al., 2024). None picks a set-valued threshold.

Thresholding for F-measure. Lewis (1995) sets F-optimal thresholds for text classification; Zhao et al. (2013) and Lipton et al. (2014) give the optimal cut on a calibrated score, which is Equation 1. Consistent plug-in estimators, including for multi-label output, follow (Ye et al., 2012; Dembczyński et al., 2011; Koyejo et al., 2014; Narasimhan et al., 2014; Parambath et al., 2014). All assume a calibrated score; our subject is what happens when the score is a sample frequency.

Consistency-based calibration. Lyu et al. (2025) derive confidence from sample consistency and report that instruction tuning makes it harder to calibrate; Huang et al. (2024) calibrate long-form generations from self-consistency. Both target one confidence score per answer. Our object is a per-candidate threshold for a set, the sharpness mechanism that moves it, and a zero-label predictor of it. Their tuning observation is consistent with ours, and we argue it is an instance of the same sharpness effect. RLHF is known to reduce output diversity (Kirk et al., 2024).

Combining set-valued outputs. Xie et al. (2023) aggregate repeated NER outputs with exactly the majority rule; Nair & Wang (2024) argue plain voting fits structured outputs poorly; Thirukovalluru et al. (2024) threshold atomic units. That intersection-like combination raises precision and union-like raises recall is long established (Màrquez et al., 1998; Florian et al., 2003). Entity-level self-consistency scores are also thresholded when LLMs self-annotate data for NER, with the threshold treated as a tuned hyperparameter (Xie et al., 2024). None connects the threshold to the F-measure literature or to calibration, or sets it without labels. Universal self-consistency (Chen et al., 2023) instead selects one whole sample by consensus; Appendix H compares such selectors with thresholding.

8 Discussion

What is new, and what is not. The optimality condition is classical and we use it as such. What the F-measure thresholding literature does not provide is evidence about the score that self-consistency actually thresholds, a sample frequency, and that is where our contributions lie: a measurement of

how that score is miscalibrated across set-valued tasks and model families, what explains the miscalibration, and a way to set the threshold without labels. The condition's role here is explanatory: it says why one half is wrong, which way to move, and when there is nothing to gain.

Why sample frequency and not likelihood. Token probabilities or verbalized confidence could score each candidate instead. We study frequency because it is the score self-consistency already computes, it needs no access to logits, and it applies unchanged to any model that can be sampled. Token probabilities and verbalized confidence are themselves miscalibrated, especially after instruction tuning (Kadavath et al., 2022; Tian et al., 2023; Xiong et al., 2024), so they would need the same correction rather than remove the need for it.

Which rule to use. With a working extractor, a single sample or majority vote is enough, and the disagreement statistic identifies these cases without labels. With a struggling extractor and no labels, the sharpness predictor sets θ; the constant $\theta = 0.25$ is nearly as good on average but falls below majority voting on the sharpest models. With twenty-five or more labels, a grid search is the simplest option and as good as anything we tested.

9 Limitations

The transfer test to a newer model generation covers six cells from one family; it establishes the direction of the effect for today's sharpest models, not a calibrated size. We evaluate open-weight models so that base and instruction-tuned checkpoints of the same model can be compared; the procedure needs only samples and applies to hosted models unchanged. Our models range from 360M to 32B parameters; the sharpest of them (Qwen3) already place their optimum above one half with a flat F_1 curve, and we expect the practical gain from tuning θ to keep shrinking as samplers sharpen, while the diagnostic value of disagreement remains. The threshold result matters when the model is fixed by cost, latency, licensing or privacy; otherwise a bigger model at $k = 1$ usually wins. We use micro-averaged F_1 with one global threshold per cell; instance-wise averaging has a different optimum, and per-example adaptive thresholds did not beat a single swept θ in our tests. As an estimator of θ with labels in hand, neither Equation 1 nor recalibration beats a grid search; the condition's value is explaining why one half is wrong and which way to move. The sharpness result spans 14 matched pairs across 6 families; the residual tuning effect after matching is indistinguishable from zero, and we claim no size for it. Several corpora repeat sentences between splits (up to 19.0% on WikiANN-ar); this affects every model and threshold equally but means absolute scores are not contamination-free.

10 Conclusion

When the output is a set, self-consistency faces a question the scalar theory of voting does not contain: where to threshold the frequencies. Practice uses one half, inherited from a setting where it was never a choice. The optimal cut is known for calibrated scores (Zhao et al., 2013; Lipton et al., 2014), but a sample frequency is not calibrated, so the convention is wrong in 192 of 246 cells, and a threshold chosen on held-out examples beats it in 189. The miscalibration is explained by sampling sharpness, to which temperature and instruction tuning are two routes, and sharpness is visible in the unlabeled samples. So the right threshold can be read off samples already drawn: a zero-label predictor beats a ten-label grid search, and the triage rule of Figure 3 says when aggregation is worth running at all. Two findings run against the usual advice: the gain is largest where extraction is failing, which makes it a diagnostic rather than free accuracy, and what aggregation buys is precision, not recall. On a working extractor voting buys little, and knowing that without labels is the point.

Reproducibility Statement

Raw generations for every cell, the scoring and analysis code, and scripts that regenerate every table and figure are in the supplementary material, so all numbers recompute without a GPU. Numbers in the text are emitted as macros by the analysis code. Setup details are in Appendix B.3.

AI USE STATEMENT

In this work, we used generative AI tools for prose editing, LaTeX boilerplate and code review. We have not used generative AI tools for experimental design, generating results, or selecting which results to report, and AI-generated research ideas are not applicable to this work. All citations were checked against the primary sources by the authors, and all AI-reviewed code was tested. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of EMNLP*, 2023.

Avi Arampatzis, Jaap Kamps, and Stephen Robertson. Where to stop reading a ranked list? Threshold optimization using truncated score distributions. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 524–531, 2009.

Yong Yi Bay and Kathleen A. Yearick. When more sampling hurts: The modal ceiling and correlation ceiling of test-time scaling. *arXiv preprint arXiv:2606.28661*, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for F-measure maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4040–4054, 2020.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems (MCS), LNCS 1857*, pp. 1–15. Springer, 2000.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. Few-NERD: A few-shot named entity recognition dataset. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Austin Feng, Marius Alonso, Ambroise Odonnat, Vasilii Feofanov, and Ievgen Redko. Optimal self-consistency for efficient reasoning with large language models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2511.12309.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of CoNLL*, pp. 168–171, 2003.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models. *arXiv preprint arXiv:2501.10868*, 2025.

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

Yukun Huang, Yixin Liu, Raghuveer Thirukovalluru, Arman Cohan, and Bhuwan Dhingra. Calibrating long-form generations from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13441–13460, 2024.

Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 216–223, 2003.

Abhyuday Jagannatha and Hong Yu. Calibrating structured output predictors for natural language processing. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2078–2092, 2020.

Martin Jansche. A maximum expected utility framework for binary sequence labeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 736–743, 2007.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024.

Oluwasanmi O. Koyejo, Nagarajan Natarajan, Pradeep K. Ravikumar, and Inderjit S. Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Volodymyr Kuleshov and Percy Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

Louisa Lam and Ching Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 27(5):553–568, 1997.

David D. Lewis. Evaluating and optimizing autonomous text classification systems. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 246–254, 1995.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pp. 225–239, 2014.

Yexiang Liu, Zekun Li, Zhi Fang, Nan Xu, Ran He, and Tieniu Tan. Rethinking the role of prompting strategies in LLM test-time scaling: A perspective of probability theory. In *Proceedings of ACL*, 2025.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Lluís Màrquez, Lluís Padró, and Horacio Rodríguez. Improving tagging performance by using voting taggers. In *Natural Language Processing and Industrial Applications*, 1998.

Inderjeet Nair and Lu Wang. MIDGARD: Self-consistency using minimum description length for structured commonsense reasoning. In *Proceedings of ACL*, 2024.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1946–1958, 2017.

Shameem A. Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing F-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6 (3):21–45, 2006.

Dietrich Rebholz-Schuhmann, Antonio Jimeno Yepes, Erik M. van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. CALBC silver standard corpus. *Journal of Bioinformatics and Computational Biology*, 8(1):163–179, 2010.

Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL*, pp. 1–8, 2004.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? A study on the impact of format restrictions on large language model performance. In *Proceedings of EMNLP: Industry Track*, 2024.

Team OLMo. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025.

Raghuveer Thirukovalluru, Yukun Huang, and Bhuwan Dhingra. Atomic self-consistency for better long form generations. In *Proceedings of EMNLP*, 2024.

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

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL*, 2003.

Harsh Verma, Sabine Bergler, and Narjesossadat Tahaei. Comparing and combining some popular NER approaches on biomedical tasks. In *Proceedings of the 22nd Workshop on Biomedical Natural Language Processing (BioNLP)*, pp. 273–279, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. CrossWeigh: Training named entity tagger from imperfect annotations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*, 2023.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot NER with ChatGPT. In *Proceedings of EMNLP*, 2023.

Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Self-improving for zero-shot named entity recognition with large language models. In *Proceedings of NAACL (Short Papers)*, pp. 583–593, 2024.

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

An Yang, Baosong Yang, Binyuan Hui, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Nan Ye, Kian Ming A. Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measures: A tale of two approaches. In *International Conference on Machine Learning (ICML)*, 2012.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699, 2002.

Ming-Jie Zhao, Narayanan Edakunni, Adam Pocock, and Gavin Brown. Beyond Fano's inequality: Bounds on the optimal F-score, BER, and cost-sensitive risk and their implications. *Journal of Machine Learning Research*, 14(32):1033–1090, 2013.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*, 2024.

Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, 2012.

A Appendix

This appendix has three jobs. Sections B–C explain how the experiments were run and the checks that guard the reported numbers. Sections D–E give the reasoning and the failed attempts behind the main results. Sections F.4–F.11 are the complete tables: the main text shows one slice of each result, and these show every cell.

B How the Experiments Were Run

B.1 Generating the samples

Every (model, dataset) pair gets one generation run. In that run we produce, for each evaluation example, one greedy continuation and twenty samples at temperature 0.7 with top-p 0.95. Every value of k we report is a prefix of that one list of twenty. This matters more than it sounds. The usual practice is to run $k = 1$ greedily and $k > 1$ with sampling, which changes two things at once, and we found the size of that confound to be as large as the effect people are arguing about (Appendix I).

Sampling stops at the first of \nText:, \nQuestion:, \nSchema:, or the token cap. We added those stop strings after discovering what happens without them; see §E.

B.2 Models

31 checkpoints across 7 model families: Qwen2.5 (0.5B/1.5B/3B/7B/14B), OLMo-2 (7B, 13B), Mistral-7B, Falcon3 (3B, 7B), Granite-3.1 (2B, 8B) and SmolLM2 (360M, 1.7B), each in base and instruction-tuned form, plus Phi-3.5-mini-instruct. Having base and instruction-tuned checkpoints of the same model lets us separate the effect of tuning from the effect of model family, which a comparison across families alone cannot do.

B.3 Tasks

Nine evaluation sets, covering five kinds of set-valued output, plus two single-answer controls.

Flat NER. CoNLL-2003 (Sang & Meulder, 2003), 4 entity types, English, 500 examples. WikiANN (Pan et al., 2017) in German, Spanish, Chinese and Arabic, 3 entity types (PER/ORG/LOC), 400 examples each, from the unimelb-nlp/wikiann release. WikiANN is silver-standard data projected from Wikipedia links, so its annotation standard differs from CoNLL's; that is why we use it, since a result that only holds under one annotation guideline is not a result about set-valued aggregation.

Fine-grained NER. Few-NERD (Ding et al., 2021), 66 fine types, English, 500 examples. The prompt enumerates all 66 types verbatim.

Relation triples. CoNLL04 (Roth & Yih, 2004), 288 examples (the full parsed test split), scored as exact (head, relation, tail) matches over five relation types.

Multi-label classification. GoEmotions (Demszky et al., 2020), 400 examples, 27 emotion labels plus neutral, all enumerated in the prompt.

Keyphrase extraction. Inspec (Hulth, 2003), 400 examples, from the memray/inspec release pinned at commit ec922df. An item is a keyphrase string, matched after whitespace normalisation and lowercasing.

Single-answer controls. TriviaQA (Joshi et al., 2017) and a synthetic JSON extraction task at 2, 5, 10, 20 and 50 fields. These produce one answer, so there is no candidate set and no threshold to place. They appear in the tables because they were part of the same sweep and because the measurement corrections apply to them, but the threshold result does not concern them.

B.4 Train/test overlap in these corpora

Few-shot exemplars come from the training split, and the pipeline asserts they are disjoint from the evaluation ids. That assertion is weaker than it looks: shot ids carry a train prefix and evaluation ids a test one, so the intersection is empty by construction and the check can never fire. It also compares identifiers rather than text. We therefore added a check on the text itself, which drops any exemplar whose sentence also appears in the evaluation set, and measured how much overlap the corpora contain in the first place:

corpus	test items also in train
CoNLL-2003	3.6%
Few-NERD	9.6%
WikiANN-ar (worst)	19.0%

This is a property of the corpora rather than of our sampling. It is identical across models, thresholds and sample budgets, so it cannot generate any of the differences this paper reports. It does mean that absolute scores on these sets are not contamination-free, and we do not present them as such.

B.5 Sampling the evaluation sets

Evaluation sets are 500 examples drawn with a seeded random generator (300 for the auxiliary JSON sweep and for the extended-budget runs). We draw randomly rather than taking the first N rows because CoNLL's test split is ordered by document, and its first few hundred sentences are dominated by sports result tables — a sample that is neither random nor representative.

Few-shot examples come from the training split, and the runner checks before generating that no evaluation example appears among them.

B.6 Scoring

Nothing is scored during generation. The raw samples are written to disk, and every number in this paper is computed afterwards from those files. This is why we could revise the parser, the metric and the aggregation rule several times without regenerating anything, and it is why a reader can recompute any of our numbers without a GPU.

For NER we compare multisets of (text, type) pairs, so a mention that appears twice in the gold annotation must be produced twice.

C Checks and Safeguards

C.1 Numbers in the text are generated, not typed

Every number that appears in the prose is written by the analysis code as a LaTeX macro and referenced by name. No number is typed by hand. Typed numbers drift from results; generated numbers cannot.

C.2 The integrity checker

The analysis refuses to run unless the store of generated samples passes seven checks:

1. every record's own dataset field matches the dataset its filename claims, so no table can quietly mix two datasets; 2. Spider prompts contain a real schema rather than only a database name; 3. every example has exactly the declared number of samples; 4. few-shot example ids never appear among the evaluation ids; 5. no example id appears twice within a cell; 6. every confidence interval we report, in every results file, contains its own point estimate; 7. evaluation-set sizes agree across the cells of a dataset, and a constrained-decoding cell covers the same examples as its unconstrained counterpart.

Each check exists because we hit the corresponding failure. Check (6) is worth copying: it is two lines of code, and it catches a failure that is otherwise easy to ship: intervals that do not contain their own estimates.

C.3 The checker is itself tested

A checker that passes everything is worthless. Ours ships with a self-test that takes the real sample store, splices two hundred Few-NERD records into the middle of a CoNLL cell, truncates another

cell's sample lists from twenty to seven, and asserts that both are caught. A checker that inspected only the first rows of each file would pass exactly that sabotage, so the test guards against it.

D THE OPTIMAL THRESHOLD IN DETAIL

D.1 DERIVING THE CONDITION

Let P be the number of items emitted at threshold θ and G the number of gold items, so that $F_1 = 2TP/(P+G)$. Lower θ slightly, so that dN more candidates are emitted, and suppose a fraction q of them are correct. Then $dTP = qdN$ and $dP = dN$, so

$$\mathrm{d}F_1 = \frac{2\,\mathrm{d}N\,\big[q(P+G) - \mathrm{TP}\big]}{(P+G)^2},$$

which is zero exactly when $q = TP/(P+G) = F_1/2$.

In words: keep admitting candidates for as long as the ones you are admitting are right more than half as often as your current score. A system at $F_1 = 0.6$ should accept a candidate whose chance of being correct is better than 0.3. Requiring a majority — better than 0.5 — is the right rule only for a system already scoring $F_1 = 1$.

D.2 WHAT THE SINGLE-ANSWER CASE DOES AND DOES NOT CONTAIN

For binary majority voting under 0–1 loss there is no threshold to choose. If each of k votes is right with probability p, the majority is right with probability $Pr[Binom(k,p) > k/2]$ (Condorcet, 1785; Lam & Suen, 1997; Kuncheva, 2004; Polikar, 2006; Zhou, 2012), and one half is the definition of majority rather than a setting.

This does not mean scalar voting has no threshold anywhere. Plurality voting over more than two alternatives contains no one half at all. Selective prediction gives scalar voting a real threshold — emit the plurality answer only when its vote share clears θ, otherwise abstain — which is the standard risk–coverage knob and is well studied. Weighted voting replaces the single knob with a vector of them. The accurate claim is narrow: the threshold in the set-valued case is not inherited from the definition of majority, and its F_1-optimal value is not one half.

D.3 RELIABILITY OF THE SAMPLE FREQUENCY

For every candidate we have a sample frequency and a correctness label, so we can bin by frequency and read off how often candidates in that bin are right. A perfectly calibrated score would put the "% correct" column on top of the "mean f" column and the gap at zero. Table 4 shows it is negative in essentially every bin of every family: the frequencies are overconfident throughout, and the gap widens as the frequency rises. The practical consequence is the last row of each block — candidates that appear in nearly every sample, which any conventional threshold admits, are the ones whose frequency overstates them most.

D.4 ONE GLOBAL THRESHOLD, FITTED ON NO LABELS

Per-cell tuning needs labels for every cell. The cheaper rival is one constant used everywhere. Scored on held-out examples over 259 cells, the best single constant is $\theta = 0.250$, worth +0.029 F_1 over the convention and better than it in 176 of 259 cells. That is 66% of what per-cell grid search buys and 54% of what a per-cell oracle buys, for no labelling effort at all. The curve is flat: every constant in $[0.100, 0.475]$ beats one half on average.

We report this because it is the natural baseline for our own practical recommendation. If you have no labels, use roughly one fifth rather than one half and you will capture most of the available gain. Per-cell tuning is a refinement on top of that, not the main event.

D.5 THE MULTI-LABEL RESULT IS PARTLY AN AVERAGING ARTIFACT

Multi-label classification is the one family where the best threshold sits above one half, and it is tempting to read that as a clean confirmation: the models over-produce labels, so the frequency is miscalibrated upward, and the condition predicts the direction while the naive plug-in rule does not.

That is not the whole story. Everything in this paper optimises micro-averaged F_1 with a single global threshold. The multi-label literature has long observed that micro, macro and instance-wise averaging have different optima (Dembczyński et al., 2011; Ye et al., 2012). We therefore recomputed the GoEmotions optimum under instance-wise averaging —F_1 computed per example and then averaged — on exactly the same candidates:

	micro	instance-wise
cells with $\theta^* > 1/2$	7 of 29	1 of 29

Under instance-wise averaging the effect largely disappears. So the accurate statement is that the above-half optimum on multi-label is specific to micro-averaging, and is at least as much a property of the averaging choice as of the model's calibration. We therefore do not treat this family as a clean confirmation of the condition. It is a reminder that "the" optimal threshold is defined only once you have said which F_1 you mean.

D.6 THE CONDITION IS NECESSARY, NOT SUFFICIENT

Equation (1) characterises a stationary point. It does not by itself identify the global optimum, because F_1 along a candidate ranking is not concave: $q(\theta) - F_1(\theta)/2$ can change sign several times, and the condition holds at every crossing. We counted the crossings. Only 114 of 246 cells have a single one; the worst has 43. Our solving rule — walk down the ranking and stop at the last point where the band precision still exceeds $F_1/2$ — therefore picks a crossing, not necessarily the best one, and this is one reason it does not beat a grid search. A reader should treat the condition as describing where an optimum must sit, not as an algorithm for finding it.

Two further scope limits we should state plainly. The result is for micro-averaged F_1 with a single global threshold and a fixed gold count; instance-wise and macro averaging have different optima, a point the multi-label literature makes at length (Dembczyński et al., 2011; Ye et al., 2012). And the perturbation dN is continuous while frequencies are multiples of $1/k$, so the condition is an approximation at small k in the same way it is for any derivative taken over a discrete grid.

D.7 WHERE THE CONDITION DOES NOT APPLY

Equation (1) describes an interior optimum: a point where admitting one more candidate stops paying. Some cells have no such point. If a model already produces exactly the gold set and nothing else, there is no marginal candidate, the best threshold is any value that keeps what it has, and the condition is vacuous rather than wrong.

That is not hypothetical. Our synthetic JSON task is set-valued in the right way — a candidate is a (field, value) pair — but the models solve it: six of its cells sit at $F_1 \geq 0.99$. We list them in Table F.7 and exclude them from the fit, because scoring a first-order condition on a corner solution would tell us nothing either way.

D.8 WHAT THE AGREEMENT IN FIGURE 2 DOES AND DOES NOT SHOW

This is the most important caveat in the paper and we would rather state it ourselves than have it found for us.

Equation (1) is a first-order condition evaluated at an argmax. We locate θ^* by sweeping the candidate ranking for the rank that maximises F_1, and then measure q as the gold rate in a band centred on that same rank. Moving from rank i to $i+1$ raises F_1 exactly when the candidate at $i+1$ is correct more often than $F_1(i)/2$, so a band average taken at the maximum is squeezed toward $F_1^*/2$ by the

definition of the maximum. Some agreement is therefore guaranteed before any data is collected. The question is how much.

We answer it with three controls, all in isr/nulls.py.

A mechanical null. We build synthetic ranked lists whose precision decays smoothly along the rank, $p = b(1 - x)^a$ with a, b, list length and gold count drawn at random. Nothing is assumed except that the ranking is informative and locally smooth; no claim about F_1 is built in. Running our own estimator on these gives MAE 0.020 and correlation 0.914, against 0.033 and 0.915 on the real data. The null fits at least as well as our measurements do.

A permutation null. Shuffling f_m within a cell keeps the frequency distribution and the gold count but destroys any link between how often a candidate is sampled and whether it is correct. Its MAE is 0.063 against our 0.033 — clearly worse, which is the one thing this family of tests establishes: sample frequency does carry information about correctness.

We must immediately qualify that, because it is the statistic and not the null that is doing the work. On correlation the permutation null scores 0.950 against our 0.915 — that is, destroying the link between frequency and correctness improves the correlation that Figure 2 plots. We report both statistics for the null, since reporting only one would be selective. The lesson is the one the rest of this section draws: correlation across cells is driven by between-cell variation in F_1 level and carries almost no evidence about the condition. MAE is the only one of the two that discriminates at all, and even it is beaten by the mechanical null.

The same statistic at non-optimal ranks. Reading the band at the 25th, 50th and 75th percentile of the ranking instead of at the argmax gives MAE 0.146, 0.084 and 0.095. So the argmax location does matter; a statistic read at an arbitrary rank is far off.

What we conclude. Figure 2 is not a test of Equation (1), and we no longer describe it as one. Equation (1) is a theorem; it does not need testing, and it could not have failed. What the figure measures is the size of the discretisation and band-width error in our own estimator — useful for judging whether the quantity is measurable at this sample size, and nothing more. A reader should take the correlation as a sanity check on our instrumentation, not as evidence for a mechanism.

The estimator is also not the one the prose describes. We speak of the candidates one is "on the verge of admitting", which are the ones just below the cut, but q is measured over a symmetric band straddling it. Measured one-sided over the 400 candidates below the cut, the fit is markedly worse: MAE 0.074 and slope 0.422, against 0.033 and 1.143 symmetric. We report the symmetric band because it is what the code has always done, and we flag the discrepancy rather than quietly adopting whichever is more flattering. The one-sided number is the more honest measure of how well a practitioner could estimate q from below the cut alone.

D.9 The band width is a free parameter

q is the fraction of correct items among the candidates straddling the cut, so it has to be measured over a band of finite width. That width is a free parameter and the fit moves with it. Measured over our cells:

band	50	100	200	400	2000
slope	0.979	0.991	1.057	1.143	1.307
MAE	0.046	0.032	0.027	0.033	0.041

One could read the monotone rise as attenuation from measurement error: noise in a regressor pulls a slope toward zero, so a wider band should move it toward one. We do not use that reading as support for the condition. The mechanical null of Section D.8, in which nothing is true except that the ranking is monotone, produces the same monotone trend for the same reason. The trend is a property of band-averaging, not evidence about Equation (1). We report a band of 400 candidates throughout, chose it before looking at the slope, and note that the slope crosses one between bands of 50 and 100 and keeps rising after; the claim that its interval covers one is therefore a statement about this band width, and we do not lean on it.

E Failure Modes and Negative Results

We report these because they cost us time, and because two of them are mistakes a reader could easily repeat.

E.1 Generating without a stop sequence

Our first runs had no stop strings. Base and weakly-tuned models answer the question and then keep going, inventing further "Text: … Output: …" blocks until they hit the token limit. Our parser's fallback path, which scrapes any JSON-looking object out of the text, then harvested entities from those invented continuations and scored them as mistakes on the current sentence. For Qwen2.5-14B base on CoNLL this affected 353 of 500 examples.

Because the invented content differs between samples, aggregation removed it. The result looked like a large, clean precision gain. It was largely an artifact of the harness. Fixing it cut the largest precision movement by roughly a factor of three and reversed the sign of one number. The lesson is narrow and practical: with a permissive parser, a missing stop sequence looks exactly like a real effect.

E.2 Ranking samples by structural validity

An appealing idea is to pick the sample that looks best formed — parses cleanly, uses only permitted labels, contains no duplicates — and use it. We implemented it and it does not work: mean $-0.016F_1$ against majority voting, with more significant losses than wins after correction. It is in Table F.6 so that nobody else has to try it.

E.3 Naming a few label types instead of all of them

Our first Few-NERD prompt named eight of the sixty-six types as examples, in the style of "a fine-grained type such as person-athlete, location-GPE, …". Between a quarter and a half of the types the models produced did not exist in the schema, and most of their predictions clustered on the eight types we happened to mention. We were measuring our prompt. Listing all sixty-six types dropped the invented-type rate from 44.8% to 12.2% on average, and to under 5% for the stronger models.

E.4 Applying a chat template twice

This is the most expensive mistake we made, and the one that best illustrates the paper's own thesis, so we give it in full.

We built prompts with the tokeniser's apply_chat_template and handed the resulting string to the inference engine. The engine tokenised it again, adding a beginning-of-sequence token to a string that already had one. For Mistral-7B-Instruct — and only for it, of the four instruction-tuned families we ran. We checked the others directly by tokenising each template with and without special tokens: Qwen, OLMo-2 and Phi produce identical sequences either way. OLMo-2's template does emit a leading <—endoftext—>, but its tokeniser does not add a second one, so it escapes the bug for a different reason than Qwen and Phi do. For Mistral — the doubled prefix pushed the model into echoing the control token [INST] before its answer. Our JSON parser takes the first balanced bracket block, found [INST], failed to parse it, and recorded the example as a total failure. The answers themselves were frequently correct. This affected 336 of 500 CoNLL prompts, 184 of 400 WikiANN-de, and 141 of 500 Few-NERD.

The effect on our own numbers was large, and all in the direction of making our result look better:

	with the bug		fixed	
cell	F_1 at $\theta{=}\frac{1}{2}$	gain	F_1 at $\theta{=}\frac{1}{2}$	gain
Mistral / CoNLL	0.188	+0.172	0.466	+0.030
Mistral / Few-NERD	0.000	+0.233	0.366	+0.004
Mistral / CoNLL04	0.000	+0.082	0.263	+0.004

Two of these cells scored exactly zero at the conventional threshold, which made the gain from moving the threshold look enormous; one of them alone contributed almost half a point of F_1 to a 24-cell mean. After the fix the cells are ordinary, and the gains are near zero. We are reporting this rather than quietly regenerating because it is the clearest evidence we have for the paper's practical claim. A large gain from moving the threshold is a signal that the candidate set is full of items the model produces rarely and gets wrong. That is usually a broken pipeline, and here it was ours. A reader who finds a large threshold gain in their own system should look upstream before celebrating.

E.5 Showing the model escape sequences instead of text

The third failure of the same family, and the most instructive, because the diagnostic pointed straight at it.

Few-shot demonstrations were built with Python's json.dumps, which defaults to ensure_ascii=True. Every non-ASCII target was therefore shown to the model as escape sequences rather than as text:

```
Output: [{"text": "\u062f\u0627\u064a\u0648 ...", "type":
"ORG"}]
```

The instruction says to copy the mention as it appears, and the demonstrations show something else, so the models copied the demonstrations. Escape sequences appeared in 63% of Arabic greedy outputs and 51% of Chinese ones, against 0% for English and 6% for Spanish. Models cannot reliably hand-encode UTF-16, so the answers came out as noise — which is exactly what we observed and misread as hallucinated spans.

Fixing it (ensure_ascii=False) and regenerating all 48 WikiANN cells:

	escaped		fixed	
cell	F_1 at $\frac{1}{2}$	gain	F_1 at $\frac{1}{2}$	gain
Qwen2.5-14B-it / WikiANN-ar	0.056	+0.071	0.583	+0.021
Qwen2.5-7B-it / WikiANN-ar	0.095	+0.110	0.479	+0.011
Phi-3.5-mini / WikiANN-ar	0.004	+0.020	0.512	+0.026
Qwen2.5-14B-it / WikiANN-zh	0.193	+0.044	0.527	+0.007

The signature is identical to the other two bugs: a cell that looks like a weak model with a large threshold gain turns out to be a working model behind a broken harness, and the gain evaporates once the harness is fixed. Arabic was not a capability result. Three for three, the diagnostic pointed at a real problem. This is the practical form of the recommendation in §5: the diagnostic tells you where to look, not what you will find.

E.6 Joining Chinese characters with spaces

WikiANN's Chinese portion is tokenised into single characters. We built the input text, and the gold spans, by joining tokens with a space, which is correct for German, Spanish and Arabic and produces nonsense for Chinese: a sentence arrived as . Models that read Chinese fluently were asked to tag text no one has ever written, and we recorded the result as a capability measurement. We now join Chinese without a separator and rebuild the gold spans the same way. We also checked Arabic, whose scores were low, and got this wrong the first time. We observed that the predictions were spans absent from the input rather than near-misses, and concluded it was genuine model failure. The observation was right and the inference was backwards, as §E.5 describes: the spans were absent from the input because the model was hand-encoding characters it had been shown as escape sequences. This is the third bug of the same family we found in our own harness, and the one our own diagnostic pointed at and we still misread.

E.7 A threshold placed on a mass point

Frequencies are multiples of $1/k$ and pile up at $f = 1$: for Phi-3.5-mini, 36% of all candidates appeared in every single sample. A rule written as $f > \theta$ with θ set to 1.0 therefore emits nothing

at all, and one of our cells scored a catastrophic -0.267 for that reason alone. (These two figures are recorded from the run that had the bug, so unlike every other number in this paper they are not regenerated by the current code, which no longer produces the failure.) We now place θ strictly between adjacent achievable frequencies.

F Complete Results

The following tables give every cell. Aggregation is majority voting at $\theta = 1/2$ unless stated otherwise, and $k = 20$ where a single k is shown.

F.1 Aggregation against scale, in detail

A paper about how to spend compute at test time owes an account of the obvious alternative: spending it on a bigger model. We count FLOPs from measured token lengths as $2P(T_{prompt} + kT_{out})$, charging the prompt once per example because the k samples share it, and compare each model at $k = 20$ against the next size up at $k = 1$(Table 5, Appendix).

The comparison goes against aggregation. In 7 of 8 pairs the larger model at one sample is both better and cheaper: twenty samples cost a median 1.4 $\times$ the FLOPs of a single sample from the next size up, and usually buy less. The exception is Few-NERD at 7B, where the fine-grained label space leaves enough removable low-frequency output for thresholding to win.

We say this plainly because it bounds everything else here. If a bigger model is available and you can afford to serve it, that is usually the better purchase. The threshold result matters in the common case where you cannot — the model is fixed by cost, latency, licensing or privacy — and the only remaining choice is what to do with samples you are already drawing. There the threshold is free, and the conventional one is leaving +0.037 F_1 on the table.

F.2 Choosing θ from few labels

We expected the labelled plug-in — Equation (1) solved on the development half — to be the more sample-efficient estimator of the two, since it needs only a local quantity where a grid search must find the argmax of a whole noisy curve. It is not: it trails grid search at every budget we tried, by +0.006 F_1 on average (Table 6). We report this because it is the obvious thing to check and the answer is not the one we guessed. The zero-label sharpness predictor of §3, by contrast, needs no development labels and beats a small-budget grid outright. Either way, tuning θ turns out to be nearly free: 10 labelled examples already recover 47% of all the gain available at any budget we measured, and a single global constant fitted on no labels captures 66% of what per-cell tuning buys (Appendix D.4).

F.3 The condition by task family

Relation extraction is the weakest fit. Its correlation is low, but so is its range: every model we tested scores between 0.000 and $0.304F_1$ on CoNLL04, so $F_1^*/2$ varies over a narrow band and a correlation taken across 26 such cells is not a stable statistic. We report it for completeness and do not interpret it. The direction is the same as everywhere else: the best threshold is below one half in 23 of 26 cells, and moving there is worth a median $+0.080F_1$.

F.4 The full grid

The complete table for this section is in the supplementary material (Section F.4 of `supplement.pdf`).

F.5 Precision and recall, every cell

The complete table for this section is in the supplementary material (Section F.5 of `supplement.pdf`).

F.6 Every aggregator

The complete table for this section is in the supplementary material (Section F.6 of `supplement.pdf`).

F.7 The optimality condition, every cell

The complete table for this section is in the supplementary material (Section F.7 of `supplement.pdf`).

F.8 The whole threshold sweep

The complete table for this section is in the supplementary material (Section F.8 of `supplement.pdf`).

F.9 Compute

The complete table for this section is in the supplementary material (Section F.9 of `supplement.pdf`).

F.10 Greedy against sampled at $k = 1$

The complete table for this section is in the supplementary material (Section F.10 of `supplement.pdf`).

F.11 Base against instruction-tuned

The complete table for this section is in the supplementary material (Section F.11 of `supplement.pdf`).

F.12 Constrained decoding

Forcing the output through a JSON schema raises format compliance on the extraction tasks — though not always to 1.00, and on the JSON-50 cells it falls, because a 50-field object does not fit the token budget and truncation is scored as a format failure.

The accuracy change should not be read as a bound on how much of the measured error can be blamed on output form, for two reasons. First, the largest deltas we quoted came from Few-NERD cells whose constrained runs predated the prompt fix that enumerates all 66 types, so they were measuring a prompt change rather than the constraint; those cells were regenerated before the numbers reported here. Second, on the cells where format compliance was already 1.00 unconstrained, forcing the grammar reduces accuracy in most cells and significantly so in a substantial minority. The accurate statement is that constrained decoding buys compliance the model was mostly already delivering, and charges accuracy for it.

F.13 Threshold transfer, per cell

θ chosen on one half of the evaluation set and applied to the other.

F.14 Threshold stability across sample budgets

θ^* re-estimated from truncations of a single 60-sample run, and the deployment test: choose θ at $k = 10$, apply it at $k = 60$.

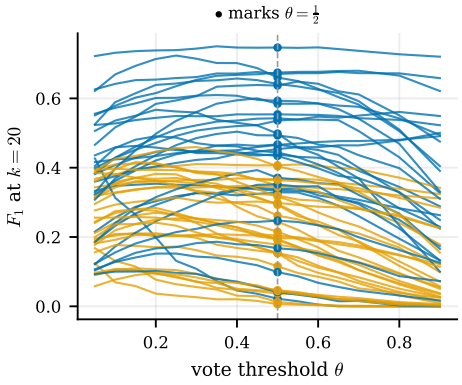


Figure 4: F_1 across vote thresholds. Blue: CoNLL-2003; orange: Few-NERD. • marks the conventional $\theta = 1/2$. The curve is broad near its maximum for strong models and sharply peaked for weak ones, which is why the cost of the convention varies so much across cells.

F.15 The shape of the frontier

G Choosing a Point on the Frontier

The threshold family of §3 is one curve in precision/recall space, and Equation (1) locates its best point for F_1. A deployment with a different loss wants a different point on the same curve: an application paying more for a false positive than for a miss should raise θ above the F_1-optimum, and one that cannot afford misses should lower it. The curve is broad near its maximum for strong models (Figure 4, Appendix), so that choice is cheap to make and cheap to get slightly wrong.

G.1 What this means for allocating compute

The mechanism supplies a procedure needing no scaling exponent. Draw a modest number of samples once, measure the precision of the candidates near the prospective cut, and

compare it against half the achieved F_1: that comparison says which way to move θ, and it is computable from the same samples that produced the output. If the oracle over those samples greatly exceeds the best threshold rule, the information is present but no threshold can reach it, and a whole-sample selector is the right instrument — consensus selection recovers part of that gap (Appendix K), and constraint scoring does not.

Most importantly, when aggregation does help, F_1 understates what changed: a +0.106 median precision movement appears as a +0.020 median F_1 movement. A pipeline whose downstream cost is dominated by false positives — populating a database, filing a claim, raising an alert — gains more from aggregation than its F_1 suggests; one that cannot afford misses gains less. Neither is visible in a single aggregate number.

H Selecting Whole Samples

The pass@k oracle — which may consult the gold label and select a whole sample — raises precision and recall together, and exceeds every threshold rule. That gap is not reachable by any choice of θ, because it requires information the frequencies do not contain.

A selector that picks one entire sample is not confined to the frontier. We evaluate two that use only the model's own outputs. Consensus selection, which picks the sample with the highest mean agreement with the others, beats majority voting in 37 of 58 cells (mean +0.023 F_1), significantly in 28 after correction against 11 significant losses; it is a form of minimum Bayes risk decoding (Chen et al., 2023). Constraint scoring, which ranks samples by structural validity alone, does not work:

mean -0.016 F_1, with 16 significant losses against 7 significant wins. We report the negative result because structural-validity heuristics are an appealing and frequently proposed remedy, and on this evidence they do not earn their place.

I THE MEASUREMENT CORRECTIONS IN FULL

Independently of where the threshold goes, three practices distort published curves for this kind of task. All three appear in work we otherwise follow. We size each; the full tables are in Appendix F.10.

Format failures are not reasoning failures. A generative extractor can fail two ways: it can emit something that is not a list of typed items at all, or it can emit a perfectly well-formed list containing a label outside the permitted set. Collapsing these into one "compliance"

number, then scoring non-compliant output as a semantic mistake, makes any decoding change that perturbs surface form look like a change in ability.

Measured apart, they behave differently. Format compliance is saturated for most cells — median 1.000, with 238 of 285 at or above 0.99 — though not uniformly: the weakest is smollm2-360m-it on GoEmotions at 0.380. Schema compliance moves on its own axis, ranging from 0.758 to 1.000 across the CoNLL cells while format compliance stays above 0.380. Constrained decoding is the natural check on this, and we report it with a caveat rather than as support: it raises format compliance, but on tasks where compliance was already saturated it costs accuracy (Appendix F.12). Format is not the bottleneck, which is exactly why charging semantic credit against format failures misattributes the loss.

Greedy $k = 1$ against sampled $k > 1$ changes two things. The usual protocol runs $k = 1$ greedily and $k > 1$ at $\tau > 0$, so the first step of the curve mixes the aggregation effect with a temperature change — and in the direction that flatters the $k = 1$ baseline, since it is drawn from a sharper distribution than everything it is compared against. Evaluating both a greedy and a $\tau = 0.7$ sample at $k = 1$ for every cell, greedy differs from sampled by up to 0.110 F_1, and the difference survives a Benjamini–Hochberg correction in 83 of 285 cells. We should be careful about the direction: of those, 71 favour greedy and 12 favour sampling, and the median gap across all cells is only +0.015. So this is a two-sided effect that is small in the typical cell and large in the tail, not a uniform thumb on the scale. What matters is that when it is large it lands on whichever arm happens to be greedy, and in the usual $k = 1$-versus-$k > 1$ comparison that is the baseline. Every k in this paper comes from one generation run at one temperature, so the sweep varies one thing.

Pooling label spaces. CoNLL-2003 has four entity types; Few-NERD has 66. They are different tasks with different error profiles, and a fit that pools them estimates a slope describing neither. Every fit here is within a fixed (model, dataset) cell, and dataset identity is carried through the pipeline to a mechanical check that rejects any table pooling them (Appendix C.2).

J COMPARING THE RULES DIRECTLY

J.1 WHY THE CONDITION IS ABOUT THE MARGINAL BAND, NOT THE THRESHOLD

There is an obvious simpler rule lurking here. If the fraction f_m of samples containing a candidate were the probability that the candidate is correct, then $q(\theta) = \theta$ and Equation (1) would collapse to $\theta^* = F_1^*/2$ — a threshold you could read straight off your score, with nothing to measure.

Sample frequency is not that probability. A mention the model produces in seventy per cent of samples is usually correct rather less than seventy per cent of the time, and how much less varies by task. Averaged over cells, the candidates at the cut are correct -0.261 less often than their frequency suggests on multi-label classification, and by smaller margins elsewhere (Table 7, Appendix).

This is worth checking rather than assuming, so we compare the two rules and the convention on identical cells:

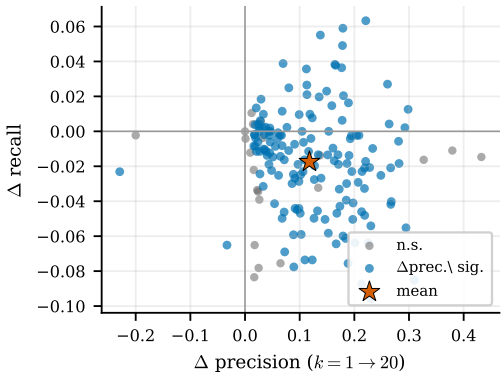
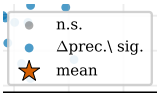


Figure 5: Aggregation buys precision, not recall. Each point is a model–dataset cell; axes are the change from a single sample ($k = 1$) to majority vote ($k = 20$). Points pile up to the right of the vertical axis (precision up) and hug the horizontal one (recall flat). The star is the mean (+0.106 precision, -0.014 recall).

	mean abs. error	correlation
$q(\theta^*) = F_1^*/2$ (ours)	0.034	0.913
$\theta^* = F_1^*/2$ (assumes calibration)	0.169	0.417
$\theta^* = 1/2$ (the convention)	0.227	–

Ours is 5.0 × more accurate than the calibrated-frequency rule, and widest where the miscalibration is worst. One caveat on how to read that table: the first row predicts a precision and the others predict a threshold, so the ratio is not a like-for-like accuracy comparison. The like-for-like comparison is downstream F_1 under each rule, which is §3's transfer exper-



iment, and there the theory rule and a plain grid search are indistinguishable. What the table does show is that assuming sample frequencies are calibrated is a bad assumption.

The same fact bears on an odd observation: on multi-label the best threshold sits above one half in 7 of 29 cells. Part of the explanation is miscalibration — the models over-produce labels, so the bar has to move up. But only part. Everything here optimises micro-averaged F_1, and micro, macro and instance-wise averaging have different optima (Dembczyński et al., 2011; Ye et al., 2012). Under instance-wise averaging only 1 of 29 remain above one half (Appendix D.5), so this family is not the clean confirmation an earlier draft claimed. We report it as a caveat.

K What Thresholding Buys, in Full

What moving the threshold buys is lopsided in a way a single F_1 number hides. Everything a threshold rule emits came from the model: lowering θ keeps more of what was produced, raising it keeps less, and nothing you do to θ can add a mention the model never produced. Recall therefore has a ceiling no threshold can lift, while precision improves by dropping candidates that appear rarely. Combining samples buys precision, and how much depends on how much of the error is low-frequency junk.

Table 11 reports precision, recall and F_1 at $k = 1$ and $k = 20$ under majority voting, with paired example-bootstrap intervals and Benjamini–Hochberg correction across the 174 cells. Precision rises in 170 of 174 cells (median $+0.106$, up to 0.432), significantly in 150 after correction, against

only 2 that move significantly the other way. Recall is flat or mildly negative (median -0.014), significantly negative in 49 cells and significantly positive in 4.

The recall movement is thus much weaker than the precision movement, and we do not describe the effect as a trade: it is a precision gain that is sometimes, not always, paid for in recall. F_1 moves by a median of $+0.020$, positive in 132 of 174 cells — so F_1 does register the effect, but it compresses a +0.106 precision movement into a +0.020 one, which is what makes aggregation look inert when only F_1 is reported. Which of the two a deployment is buying matters whenever the cost of a false positive differs from the cost of a miss.

Entity union, the $\theta \to 0$ end of the family, behaves as the monotonicity argument requires: recall up and precision down in every cell. We report it as a sanity check on the frontier rather than as a finding.

K.1 Where the headroom is

The pass@k oracle — which may consult the gold label and pick a whole sample — raises precision and recall together and exceeds every threshold rule. That gap is not reachable by any θ, because it needs information the frequencies do not contain. Selectors that pick one entire sample are not confined to the frontier: consensus selection beats majority voting in 37 of 58 cells, while ranking samples by structural validity alone does not work at all. Both are reported in Appendix H.

L Sensitivity of the Fit

How sensitive is the fit to these choices? Enough that we would rather state it than be found out. Restricting to better cells steepens the slope monotonically (1.148 on the full set; it rises above 1.5 if we keep only cells scoring above 0.4), and dropping the WikiANN family moves it below one. Combined with the band-width sensitivity of Appendix D.9, the slope ranges roughly 0.9–1.7 across defensible choices. This is the main reason we do not treat the slope as evidence for anything, and report the plug-in comparison of §4 instead, which is a held-out prediction rather than a fitted line.

M Limitations in Full

A bigger model is usually the better buy. The obvious alternative to spending compute on samples is spending it on a larger model. Counting FLOPs from measured token lengths and comparing each model at $k = 20$ against the next size up at $k = 1$, the larger model at one sample is both better and cheaper in 7 of 8 pairs: twenty samples cost a median $1.4\times$ the FLOPs of a single sample from the next size up and usually buy less (Appendix F.1). This bounds everything else here. If a bigger model is available and you can serve it, that is usually the right purchase. The threshold result matters in the common case where the model is fixed — by cost, latency, licensing or privacy — and the only remaining choice is what to do with samples you are already drawing.

What the derivation is, and what it is not. Equation (1) is a deterministic identity over realised confusion counts. It makes no probabilistic assumption — and therefore no probabilistic claim. It would be a category error to present "it does not assume independence" as an advantage over the binomial analysis of voting: Condorcet's theorem predicts an out-of-sample accuracy and can be wrong, whereas a first-order condition on an observed curve cannot be. What the condition does require is that the band around the cut be estimated well enough to locate the optimum, and Appendix D.8 is candid about how much of the apparent agreement that buys us.

Our claim that the threshold has no scalar analogue is also narrower than it first appeared. It holds for binary majority voting under 0–1 loss, where one half is the definition rather than a setting. Plurality voting over more than two alternatives has no one half either, and selective prediction gives scalar voting a genuine threshold in the form of a risk–coverage knob.

Scope. Frequencies are multiples of $1/k$ and mass at $f = 1$, so θ^* is identified only up to the gap between adjacent achievable values. Everything here is micro-averaged F_1 with one global threshold; instance-wise and macro averaging have different optima, which matters for multi-label (Appendix D.5). We study models up to 14B parameters, and the closed form is specific to the more-than-half rule — verifier-based selection (Cobbe et al., 2021; Lightman et al., 2024) and tree

search (Yao et al., 2023) are not subject to it and can exceed our curves, with the pass@k column bounding by how much.

Metrics and tasks. NER is scored at the mention level over (text, type) multisets rather than token spans, because generative models do not emit offsets; this is standard for generative NER but is not CoNLL span scoring. Text-to-SQL uses Spider's exact set match,

which ignores execution equivalence. CoNLL-2003 test labels carry known annotation errors (Wang et al., 2019), which bound achievable F_1 but affect every k equally. The synthetic JSON task is deliberately easy and anchors the saturated end of the range.

N THE CALIBRATION GAP CONTROLLED FOR ACCURACY

Expected calibration error is not invariant to accuracy, so a model that is simply worse at the task can look worse calibrated for reasons unrelated to how its vote share behaves. Controlling for accuracy does not shrink the tuned/base gap. Regressing ECE on the tuned/base indicator with dataset fixed effects gives a coefficient of +0.101; adding F_1 at $\theta = 1/2$ as a covariate moves it to +0.102 (s.e. 0.007) — slightly larger, not smaller. Pairing each tuned cell with the base cell closest in F_1 on the same dataset gives a mean gap of +0.110 over 47 pairs, with the tuned model worse in 46 of them.

O THE TEMPERATURE SWEEP

14 matched base/instruct checkpoint pairs, six datasets, five sampling temperatures. Each entry averages over the cells in that arm. Calibration error falls monotonically as the sampling distribution is widened, in both arms, and the optimal threshold falls with it — which is the same movement instruction tuning produces, by the same route.

O.1 MODERN (2025) MODELS

Two 2025 instruction-tuned checkpoints on three datasets at $k = 20$, added to probe past the grid's 14B / 2024 ceiling. They are the sharpest models we measured, and their optimal thresholds are correspondingly high — the sharpness mechanism of §4 taken to its limit — with the F_1 curve flat enough near the top that the gain over $\theta = 1/2$ stays under +0.030 F_1 on every cell. Generated with Qwen3's thinking mode disabled so the extraction fits the token budget.

P THE RECALIBRATION LADDER

We first asked whether recalibration beats a grid search when labels are scarce, on the theory that a monotone isotonic fit generalises better than the argmax of a noisy F_1 curve. It does not: at a ten-example budget the grid search gains +0.018 F_1 against recalibration's -0.014, and grid search stays ahead at every budget up to 200 (where it is +0.039). At small samples the isotonic fit and its fixed-point solver are themselves noisy. Recalibration is an explanation of what the sweep finds, not a cheaper substitute for it.

The zero-label predictor is worth ten to twenty-five labels, and used as a prior it beats a grid search outright. The right comparison is not recalibration but our zero-label predictor (§3), which spends no labels at all. On a common held-out split — the same test half scores every estimator, gain measured against $\theta = 1/2$, 20 random dev/test draws per cell — the zero-label predictor gains +0.031 F_1 regardless of budget, while a grid search must climb: at a ten-label budget the grid gains only +0.018, so the predictor beats it by +0.013 (95% bootstrap CI $[+0.011, +0.016]$, excluding zero), and the two draw even at about twenty-five labels (+0.003). A practitioner with fewer than roughly two dozen labelled sentences is better off ignoring them and reading the threshold off the unlabelled sampling distribution.

Better still, the labels need not be discarded once they arrive. Treating the predictor as a prior mean and the grid estimate as data, the shrinkage rule $\hat{\theta} = w\theta_{grid} + (1 - w)\theta_{pred}$ with $w = b/(b + \kappa)$ — where for each cell κ is set to the value ($\kappa = 25$ for every cell here) that is best on all other cells, a genuine leave-one-cell-out choice, and the result is insensitive to it across $\kappa \in [10, 100]$ — beats the

grid search at the budgets where labels are scarcest and ties it once they are not. It gains +0.015 F_1 over grid at ten labels (CI $[+0.013, +0.017]$), stays significantly ahead through fifty labels (+0.006 at twenty-five), and converges to the grid at a hundred or more, where the prior washes out and the difference

is indistinguishable from zero in either direction. It is thus never appreciably worse than the baseline a practitioner would otherwise use, and materially better when labels are few. The effect is small in absolute F_1, as it must be when many cells have a strong extractor and little to gain, and concentrates on the weaker cells where tuning matters. This turns the negative result above into a usable one: the label-free prediction is not merely a fallback for the no-label case but makes label-based tuning itself more sample-efficient.

The predictor transfers to task families it has never seen. The concern with any fitted predictor is that it has learned the particular families in our grid. We test this directly by refitting it leave-one-family-out: to predict θ for a keyphrase cell the linear map is trained only on the NER, relation and multilabel cells, and likewise for each of the 4 families, so the predictor is always evaluated on a task type absent from its training set. The result barely moves. Overall the zero-label predictor still beats a ten-label grid ($+0.012$, CI $[+0.009, +0.015]$) and the shrinkage rule still dominates it ($+0.015$, CI $[+0.013, +0.017]$), and the shrinkage win holds separately in every one of the 4 held-out families, each with a bootstrap CI excluding zero (from $+0.003F_1$ on keyphrases to $+0.017$ on multilabel). The mapping from unlabeled sampling sharpness to the right threshold is therefore a property of the sampling distribution, not of the task: it is learned on entity recognition and relation extraction and applies, untouched, to multilabel classification and keyphrase extraction.

Is a single threshold enough, or should it adapt per example? A reader might expect that choosing θ per example from that example's own vote distribution would beat one global cut. It does not, and the ceiling is informative: even a greedy per-example oracle — which, for each of the 246 evaluated cells' examples, is handed the gold labels and picks the grid threshold that maximises that example's own F_1 — scores -0.021 F_1 below the single best global θ, and the global cut wins on 63% of cells. Locally optimal thresholds do not compose under micro-pooling: a cut that is right for one example's counts moves the pooled precision/recall balance the wrong way for the rest. Because even the oracle cannot win, no label-free adaptive rule can either. The object worth selecting is a single global θ, which is exactly what the rest of the paper does.

Does fixing the calibration fix the rule? This is the test that makes the diagnosis worth making. We fit isotonic regression (Zadrozny & Elkan, 2002) of correctness on frequency using the dev half of the examples, apply the textbook plug-in rule to the recalibrated score — emit iff $r(f) > F_1/2$, solved as a fixed point — and score it on the held-out half, against the same rule applied to the raw frequency, our marginal-band rule, and a per-cell grid search:

rule (all held out, 259 cells)	gain over $\theta = 1/2$
plug-in on the raw frequency, $\theta = F_1^*/2$	$+0.028$
marginal band, Equation (1)	$+0.041$
plug-in on the recalibrated score	$+0.044$
per-cell grid search	$+0.044$

Recalibrating the score closes essentially all of the gap between the raw plug-in rule and a grid search: the two become statistically indistinguishable (the mean difference is a fraction of an F_1 point and reverses sign across cells). We read that as convergence, not as one rule beating the other, and a practitioner with labels should still simply sweep. What it does show is why the plug-in rule fails and the sweep is necessary — the score being thresholded is not a calibrated probability — which is the question the condition was meant to answer. This is a diagnosis, not a better estimator, and we do not claim otherwise.

Table 4: Reliability of sample frequency as a predictor of correctness, pooled over models within each task family. "gap" is % correct minus mean f; negative means overconfident.

Family	bin	candidates	mean f	% correct	gap
Multi-label	0.0–0.1	18578	0.050	0.057	+0.007
	0.1–0.2	13067	0.119	0.083	-0.036
	0.2–0.3	6129	0.222	0.114	-0.108
	0.3–0.4	3987	0.323	0.137	-0.186
	0.4–0.5	2978	0.423	0.144	-0.279
	0.5–0.6	2321	0.523	0.166	-0.357
	0.6–0.7	1951	0.624	0.183	-0.441
	0.7–0.8	1736	0.725	0.202	-0.523
	0.8–0.9	1722	0.825	0.233	-0.592
	0.9–1.0	5668	0.976	0.379	-0.597
ECE 0.169					
NER (English)	0.0–0.1	191106	0.050	0.020	-0.030
	0.1–0.2	74397	0.117	0.064	-0.053
	0.2–0.3	27737	0.221	0.114	-0.107
	0.3–0.4	15813	0.322	0.166	-0.156
	0.4–0.5	10644	0.423	0.205	-0.218
	0.5–0.6	7967	0.523	0.252	-0.271
	0.6–0.7	6234	0.624	0.309	-0.315
	0.7–0.8	5434	0.724	0.370	-0.354
	0.8–0.9	5028	0.824	0.409	-0.415
	0.9–1.0	15276	0.976	0.593	-0.383
ECE 0.087					
NER (4 languages)	0.0–0.1	77764	0.050	0.036	-0.014
	0.1–0.2	36587	0.118	0.101	-0.017
	0.2–0.3	15635	0.222	0.169	-0.053
	0.3–0.4	9757	0.323	0.219	-0.103
	0.4–0.5	7276	0.423	0.277	-0.146
	0.5–0.6	5901	0.524	0.313	-0.211
	0.6–0.7	5089	0.624	0.375	-0.249
	0.7–0.8	4773	0.725	0.418	-0.307
	0.8–0.9	4972	0.826	0.476	-0.349
	0.9–1.0	21620	0.982	0.703	-0.279
ECE 0.086					
Relation triples	0.0–0.1	34120	0.050	0.025	-0.025
	0.1–0.2	13148	0.116	0.065	-0.051
	0.2–0.3	4321	0.220	0.114	-0.106
	0.3–0.4	2139	0.322	0.149	-0.173
	0.4–0.5	1376	0.421	0.182	-0.239
	0.5–0.6	971	0.522	0.237	-0.286
	0.6–0.7	717	0.626	0.305	-0.320
	0.7–0.8	524	0.724	0.323	-0.401
	0.8–0.9	438	0.823	0.393	-0.430
	0.9–1.0	1104	0.970	0.418	-0.552
ECE 0.071					
Keyphrase	0.0–0.1	199291	0.050	0.042	-0.008
	0.1–0.2	91001	0.118	0.120	+0.002
	0.2–0.3	39039	0.222	0.199	-0.022
	0.3–0.4	24622	0.323	0.269	-0.053
	0.4–0.5	17596	0.423	0.325	-0.099
	0.5–0.6	13693	0.523	0.367	-0.157
	0.6–0.7	11362	0.625	0.407	-0.217
	0.7–0.8	9979	0.725	0.450	-0.275
	0.8–0.9	9291	0.825	0.495	-0.330
	0.9–1.0	23898	0.969	0.608	-0.360
ECE 0.056					

Table 5: Compute-matched comparison: a model at $k = 20$ against the next size up at $k = 1$, with FLOPs counted from measured token lengths.

Dataset	Smaller model	smaller, k=20		next size, k=1		Cheaper
		F_1	TFLOPs	F_1	TFLOPs	& better
CoNLL	0.5b	0.371	1.1	0.386	0.8	scale
Few-NERD	0.5b	0.036	0.9	0.156	0.9	scale
CoNLL	1.5b	0.434	2.4	0.540	1.6	scale
Few-NERD	1.5b	0.178	3.2	0.214	1.7	scale
CoNLL	3b	0.554	4.6	0.649	3.8	scale
Few-NERD	3b	0.204	5.4	0.370	4.1	scale
CoNLL	7b	0.670	11.8	0.744	8.1	scale
Few-NERD	7b	0.364	14.4	0.336	8.3	sample

Table 6: How many labelled examples it takes to choose θ. Mean held-out F_1 over NER cells, 40 resamples per budget.

Labelled examples	$\theta = \frac{1}{2}$	grid search	Eq. (1)
10	0.397	0.416	0.402
25	0.397	0.426	0.416
50	0.397	0.432	0.426
100	0.398	0.435	0.433
200	0.399	0.438	0.436
400	0.396	0.436	0.433

Table 7: Equation (1) broken down by kind of set-valued output. "naive MAE" is the error of the simpler rule $\theta^* = F_1^*/2$, which assumes a sample frequency is a calibrated probability of correctness.

Task family	n	corr	MAE	slope	naive MAE	$\theta^* < \frac{1}{2}$	median gain
NER (English)	54	0.970	0.020	1.109	0.175	42/54	+0.029
NER (4 languages)	111	0.869	0.047	1.176	0.177	81/111	+0.019
Relation triples	26	0.751	0.028	1.227	0.141	23/26	+0.080
Multi-label	26	0.832	0.024	1.300	0.275	17/26	+0.011
Keyphrase	29	0.676	0.020	0.943	0.054	29/29	+0.065
All	246	0.913	0.034		0.169		

Table 8: Constrained against unconstrained decoding, same examples and seeds. FC is format compliance. The JSON-50 rows showing FC $= 0.00$ beside $F_1 = 1.000$ are not a contradiction: a 50-field object does not fit the token budget, so no individual sample parses to completion and FC — which requires every aggregated sample to parse — is zero, while the field-wise majority still reassembles every gold field from the partial samples. It is a measurement of the token cap, not of the model, and it is the clearest illustration of why we separate format compliance from accuracy.

Dataset	Model	k	FC uncon.	FC con.	uncon.	con.	Δ (95% CI)
CoNLL	0.5B	1	1.00	1.00	0.237	0.226	-0.011 [-0.031, 0.008]
CoNLL	0.5B	5	1.00	1.00	0.347	0.342	-0.005 [-0.028, 0.014]
CoNLL	0.5B	20	1.00	1.00	0.371	0.371	+0.001 [-0.026, 0.032]
CoNLL	1.5B	1	0.93	1.00	0.386	0.382	-0.004 [-0.042, 0.035]
CoNLL	1.5B	5	0.79	1.00	0.414	0.431	+0.017 [-0.012, 0.046]
CoNLL	1.5B	20	0.57	1.00	0.434	0.464	+0.030 [-0.001, 0.069]
CoNLL	3B	1	1.00	1.00	0.540	0.506	-0.034 [-0.064, -0.004]
CoNLL	3B	5	1.00	1.00	0.557	0.531	-0.026 [-0.051, -0.001]
CoNLL	3B	20	1.00	1.00	0.554	0.546	-0.008 [-0.036, 0.021]
CoNLL	7B	1	1.00	1.00	0.649	0.650	+0.001 [-0.014, 0.016]
CoNLL	7B	5	1.00	1.00	0.671	0.657	-0.014 [-0.028, -0.000]
CoNLL	7B	20	1.00	1.00	0.670	0.666	-0.004 [-0.015, 0.007]
CoNLL	14B	1	1.00	1.00	0.744	0.732	-0.012 [-0.030, 0.004]
CoNLL	14B	5	1.00	1.00	0.743	0.735	-0.008 [-0.024, 0.006]
CoNLL	14B	20	1.00	1.00	0.747	0.738	-0.009 [-0.023, 0.004]
Few-NERD	0.5B	1	1.00	1.00	0.058	0.073	+0.015 [-0.003, 0.032]
Few-NERD	0.5B	5	1.00	0.99	0.047	0.049	+0.002 [-0.015, 0.022]
Few-NERD	0.5B	20	0.99	0.99	0.036	0.026	-0.011 [-0.020, -0.003]
Few-NERD	1.5B	1	1.00	1.00	0.156	0.163	+0.007 [-0.004, 0.018]
Few-NERD	1.5B	5	0.99	1.00	0.169	0.173	+0.004 [-0.005, 0.013]
Few-NERD	1.5B	20	0.97	1.00	0.178	0.176	-0.002 [-0.013, 0.008]
Few-NERD	3B	1	1.00	1.00	0.214	0.219	+0.005 [-0.021, 0.034]
Few-NERD	3B	5	1.00	1.00	0.212	0.227	+0.015 [-0.012, 0.041]
Few-NERD	3B	20	0.99	1.00	0.204	0.216	+0.013 [-0.009, 0.035]
Few-NERD	7B	1	1.00	1.00	0.370	0.368	-0.002 [-0.012, 0.007]
Few-NERD	7B	5	1.00	1.00	0.375	0.371	-0.003 [-0.013, 0.006]
Few-NERD	7B	20	1.00	1.00	0.364	0.368	+0.004 [-0.004, 0.012]
Few-NERD	14B	1	1.00	1.00	0.336	0.295	-0.041 [-0.065, -0.020]
Few-NERD	14B	5	1.00	1.00	0.323	0.301	-0.022 [-0.042, -0.004]
Few-NERD	14B	20	1.00	1.00	0.320	0.292	-0.028 [-0.045, -0.012]
JSON-10	0.5B	1	1.00	1.00	0.995	0.994	-0.000 [-0.001, 0.000]
JSON-10	0.5B	5	1.00	1.00	0.999	0.999	-0.000 [-0.001, 0.000]
JSON-10	0.5B	20	1.00	1.00	1.000	1.000	-0.000 [-0.001, 0.000]
JSON-10	1.5B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	1.5B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	1.5B	20	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	3B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	3B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	3B	20	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	7B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	7B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	7B	20	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	14B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	14B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-10	14B	20	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	0.5B	1	0.00	0.00	0.000	0.000	+0.000 [0.000, 0.000]
JSON-50	0.5B	5	0.00	0.00	0.000	0.000	+0.000 [0.000, 0.000]
JSON-50	0.5B	20	0.00	0.00	0.000	0.000	+0.000 [0.000, 0.000]
JSON-50	1.5B	1	0.31	0.33	0.473	0.498	+0.034 [0.015, 0.058]
JSON-50	1.5B	5	0.00	0.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	1.5B	20	0.00	0.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	3B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	3B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	3B	20	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	7B	1	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	7B	5	1.00	1.00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	7B	20	1.00	00	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	14B	1	1.00	00	1.000	1.000	+0.000 [0.000, 0.000]

Table 8 (continued)

JSON-50	14B	5	0.98	0.97	1.000	1.000	+0.000 [0.000, 0.000]
JSON-50	14B	20	0.66	0.64	1.000	1.000	+0.000 [0.000, 0.000]

Table 9: Held-out threshold transfer.

Dataset	Model	$\theta{=}\frac{1}{2}$	dev-chosen	theory	oracle	Δ (theory)
CoNLL	0.5B	0.351	0.377	0.371	0.385	+0.021
CoNLL	1.5B	0.434	0.450	0.453	0.458	+0.019
CoNLL	3B	0.556	0.551	0.537	0.560	-0.018
CoNLL	7B	0.668	0.668	0.671	0.676	+0.002
CoNLL	14B	0.743	0.740	0.744	0.748	+0.001
CoNLL	0.5B base	0.364	0.356	0.291	0.367	-0.073
CoNLL	7B base	0.645	0.652	0.645	0.661	-0.000
CoNLL	14B base	0.656	0.720	0.711	0.720	+0.055
CoNLL	OLMo-2 7B base	0.462	0.492	0.498	0.508	+0.036
CoNLL	OLMo-2 7B it	0.508	0.566	0.562	0.577	+0.054
CoNLL	Phi-3.5-mini	0.452	0.515	0.498	0.515	+0.046
CoNLL	Mistral-7B it	0.463	0.487	0.483	0.490	+0.020
CoNLL	falcon3-3b-base	0.345	0.340	0.332	0.357	-0.013
CoNLL	falcon3-3b-it	0.333	0.348	0.344	0.350	+0.012
CoNLL	falcon3-7b-base	0.446	0.444	0.441	0.462	-0.005
CoNLL	falcon3-7b-it	0.588	0.595	0.605	0.608	+0.017
CoNLL	granite3.1-2b-base	0.334	0.328	0.326	0.341	-0.007
CoNLL	granite3.1-2b-it	0.539	0.561	0.548	0.568	+0.009
CoNLL	granite3.1-8b-base	0.544	0.532	0.542	0.550	-0.002
CoNLL	granite3.1-8b-it	0.673	0.674	0.675	0.679	+0.002
CoNLL	mistral-7b-base	0.465	0.459	0.459	0.470	-0.005
CoNLL	olmo2-13b-base	0.645	0.657	0.645	0.657	+0.000
CoNLL	qwen2.5-1.5b-base	0.488	0.497	0.495	0.502	+0.008
CoNLL	qwen2.5-3b-base	0.585	0.604	0.601	0.605	+0.017
CoNLL	smollm2-1.7b-base	0.252	0.243	0.244	0.253	-0.008
CoNLL	smollm2-1.7b-it	0.106	0.153	0.134	0.156	+0.028
CoNLL	smollm2-360m-base	0.178	0.188	0.186	0.194	+0.008
CoNLL	smollm2-360m-it	0.039	0.102	0.060	0.105	+0.021
Few-NERD	0.5B	0.039	0.111	0.076	0.113	+0.038
Few-NERD	1.5B	0.179	0.201	0.190	0.208	+0.011
Few-NERD	3B	0.204	0.255	0.248	0.261	+0.044
Few-NERD	7B	0.364	0.386	0.390	0.397	+0.026
Few-NERD	14B	0.327	0.390	0.393	0.394	+0.066
Few-NERD	0.5B base	0.050	0.101	0.077	0.103	+0.027
Few-NERD	7B base	0.313	0.409	0.412	0.415	+0.099
Few-NERD	14B base	0.401	0.459	0.451	0.460	+0.051
Few-NERD	OLMo-2 7B base	0.149	0.204	0.204	0.211	+0.055
Few-NERD	Phi-3.5-mini	0.351	0.348	0.348	0.357	-0.003
Few-NERD	Mistral-7B it	0.364	0.359	0.363	0.372	-0.001
Few-NERD	falcon3-3b-base	0.188	0.265	0.265	0.269	+0.077
Few-NERD	falcon3-3b-it	0.200	0.205	0.201	0.215	+0.001
Few-NERD	falcon3-7b-base	0.279	0.368	0.368	0.374	+0.089
Few-NERD	falcon3-7b-it	0.338	0.401	0.404	0.408	+0.066
Few-NERD	granite3.1-2b-base	0.148	0.286	0.283	0.287	+0.135
Few-NERD	granite3.1-2b-it	0.286	0.301	0.294	0.308	+0.008
Few-NERD	granite3.1-8b-base	0.297	0.360	0.362	0.371	+0.065
Few-NERD	granite3.1-8b-it	0.404	0.404	0.406	0.410	+0.003
Few-NERD	mistral-7b-base	0.237	0.351	0.355	0.361	+0.118
Few-NERD	olmo2-13b-base	0.256	0.337	0.333	0.339	+0.077
Few-NERD	qwen2.5-1.5b-base	0.123	0.170	0.165	0.178	+0.043
Few-NERD	qwen2.5-3b-base	0.198	0.284	0.281	0.288	+0.083
Few-NERD	smollm2-1.7b-base	0.156	0.242	0.240	0.246	+0.084
Few-NERD	smollm2-1.7b-it	0.207	0.247	0.253	0.256	+0.046
Few-NERD	smollm2-360m-base	0.051	0.112	0.055	0.118	+0.004
CoNLL04	0.5B	0.051	0.136	0.140	0.148	+0.090
GoEmotions	0.5B	0.120	0.115	0.102	0.122	-0.018
Inspec	0.5B	0.147	0.365	0.354	0.365	+0.208
WikiANN-ar	0.5B	0.099	0.225	0.239	0.247	+0.141

Table 9 (continued)

CoNLL	0.5B base	0.364	0.356	0.291	0.367	-0.073
CoNLL	7B base	0.645	0.652	0.645	0.661	-0.000
CoNLL	14B base	0.656	0.720	0.711	0.720	+0.055
CoNLL	OLMo-2 7B base	0.462	0.492	0.498	0.508	+0.036
CoNLL	OLMo-2 7B it	0.508	0.566	0.562	0.577	+0.054
CoNLL	Phi-3.5-mini	0.452	0.515	0.498	0.515	+0.046
CoNLL	Mistral-7B it	0.463	0.487	0.483	0.490	+0.020
CoNLL	falcon3-3b-base	0.345	0.340	0.332	0.357	-0.013
CoNLL	falcon3-3b-it	0.333	0.348	0.344	0.350	+0.012
CoNLL	falcon3-7b-base	0.446	0.444	0.441	0.462	-0.005
CoNLL	falcon3-7b-it	0.588	0.595	0.605	0.608	+0.017
CoNLL	granite3.1-2b-base	0.334	0.328	0.326	0.341	-0.007
CoNLL	granite3.1-2b-it	0.539	0.561	0.548	0.568	+0.009
CoNLL	granite3.1-8b-base	0.544	0.532	0.542	0.550	-0.002
CoNLL	granite3.1-8b-it	0.673	0.674	0.675	0.679	+0.002
CoNLL	mistral-7b-base	0.465	0.459	0.459	0.470	-0.005
CoNLL	olmo2-13b-base	0.645	0.657	0.645	0.657	+0.000
CoNLL	qwen2.5-1.5b-base	0.488	0.497	0.495	0.502	+0.008
CoNLL	qwen2.5-3b-base	0.585	0.604	0.601	0.605	+0.017
CoNLL	smollm2-1.7b-base	0.252	0.243	0.244	0.253	-0.008
CoNLL	smollm2-1.7b-it	0.106	0.153	0.134	0.156	+0.028
CoNLL	smollm2-360m-base	0.178	0.188	0.186	0.194	+0.008
CoNLL	smollm2-360m-it	0.039	0.102	0.060	0.105	+0.021
Few-NERD	0.5B	0.039	0.111	0.076	0.113	+0.038
Few-NERD	1.5B	0.179	0.201	0.190	0.208	+0.011
Few-NERD	3B	0.204	0.255	0.248	0.261	+0.044
Few-NERD	7B	0.364	0.386	0.390	0.397	+0.026
Few-NERD	14B	0.327	0.390	0.393	0.394	+0.066
Few-NERD	0.5B base	0.050	0.101	0.077	0.103	+0.027
Few-NERD	7B base	0.313	0.409	0.412	0.415	+0.099
Few-NERD	14B base	0.401	0.459	0.451	0.460	+0.051
Few-NERD	OLMo-2 7B base	0.149	0.204	0.204	0.211	+0.055
Few-NERD	Phi-3.5-mini	0.351	0.348	0.348	0.357	-0.003
Few-NERD	Mistral-7B it	0.364	0.359	0.363	0.372	-0.001
Few-NERD	falcon3-3b-base	0.188	0.265	0.265	0.269	+0.077
Few-NERD	falcon3-3b-it	0.200	0.205	0.201	0.215	+0.001
Few-NERD	falcon3-7b-base	0.279	0.368	0.368	0.374	+0.089
Few-NERD	falcon3-7b-it	0.338	0.401	0.404	0.408	+0.066
Few-NERD	granite3.1-2b-base	0.148	0.286	0.283	0.287	+0.135
Few-NERD	granite3.1-2b-it	0.286	0.301	0.294	0.308	+0.008
Few-NERD	granite3.1-8b-base	0.297	0.360	0.362	0.371	+0.065
Few-NERD	granite3.1-8b-it	0.404	0.404	0.406	0.410	+0.003
Few-NERD	mistral-7b-base	0.237	0.351	0.355	0.361	+0.118
Few-NERD	olmo2-13b-base	0.256	0.337	0.333	0.339	+0.077
Few-NERD	qwen2.5-1.5b-base	0.123	0.170	0.165	0.178	+0.043
Few-NERD	qwen2.5-3b-base	0.198	0.284	0.281	0.288	+0.083
Few-NERD	smollm2-1.7b-base	0.156	0.242	0.240	0.246	+0.084
Few-NERD	smollm2-1.7b-it	0.207	0.247	0.253	0.256	+0.046
Few-NERD	smollm2-360m-base	0.051	0.112	0.055	0.118	+0.004
CoNLL04	0.5B	0.051	0.136	0.140	0.148	+0.090
GoEmotions	0.5B	0.120	0.115	0.102	0.122	-0.018
Inspec	0.5B	0.147	0.365	0.354	0.365	+0.208
WikiANN-ar	0.5B	0.099	0.225	0.239	0.247	+0.141
WikiANN-de	0.5B	0.389	0.451	0.458	0.460	+0.069
WikiANN-es	0.5B	0.418	0.450	0.459	0.466	+0.041
WikiANN-zh	0.5B	0.203	0.290	0.300	0.307	+0.097
CoNLL04	1.5B	0.153	0.202	0.193	0.214	+0.040
GoEmotions	1.5B	0.185	0.230	0.220	0.231	+0.035
Inspec	1.5B	0.398	0.433	0.431	0.436	+0.033
WikiANN-ar	1.5B	0.255	0.331	0.336	0.341	+0.081
WikiANN-de	1.5B	0.364	0.462	0.462	0.467	+0.098

Table 9 (continued)

CoNLL	0.5B base	0.364	0.356	0.291	0.367	-0.073
CoNLL	7B base	0.645	0.652	0.645	0.661	-0.000
CoNLL	14B base	0.656	0.720	0.711	0.720	+0.055
CoNLL	OLMo-2 7B base	0.462	0.492	0.498	0.508	+0.036
CoNLL	OLMo-2 7B it	0.508	0.566	0.562	0.577	+0.054
CoNLL	Phi-3.5-mini	0.452	0.515	0.498	0.515	+0.046
CoNLL	Mistral-7B it	0.463	0.487	0.483	0.490	+0.020
CoNLL	falcon3-3b-base	0.345	0.340	0.332	0.357	-0.013
CoNLL	falcon3-3b-it	0.333	0.348	0.344	0.350	+0.012
CoNLL	falcon3-7b-base	0.446	0.444	0.441	0.462	-0.005
CoNLL	falcon3-7b-it	0.588	0.595	0.605	0.608	+0.017
CoNLL	granite3.1-2b-base	0.334	0.328	0.326	0.341	-0.007
CoNLL	granite3.1-2b-it	0.539	0.561	0.548	0.568	+0.009
CoNLL	granite3.1-8b-base	0.544	0.532	0.542	0.550	-0.002
CoNLL	granite3.1-8b-it	0.673	0.674	0.675	0.679	+0.002
CoNLL	mistral-7b-base	0.465	0.459	0.459	0.470	-0.005
CoNLL	olmo2-13b-base	0.645	0.657	0.645	0.657	+0.000
CoNLL	qwen2.5-1.5b-base	0.488	0.497	0.495	0.502	+0.008
CoNLL	qwen2.5-3b-base	0.585	0.604	0.601	0.605	+0.017
CoNLL	smollm2-1.7b-base	0.252	0.243	0.244	0.253	-0.008
CoNLL	smollm2-1.7b-it	0.106	0.153	0.134	0.156	+0.028
CoNLL	smollm2-360m-base	0.178	0.188	0.186	0.194	+0.008
CoNLL	smollm2-360m-it	0.039	0.102	0.060	0.105	+0.021
Few-NERD	0.5B	0.039	0.111	0.076	0.113	+0.038
Few-NERD	1.5B	0.179	0.201	0.190	0.208	+0.011
Few-NERD	3B	0.204	0.255	0.248	0.261	+0.044
Few-NERD	7B	0.364	0.386	0.390	0.397	+0.026
Few-NERD	14B	0.327	0.390	0.393	0.394	+0.066
Few-NERD	0.5B base	0.050	0.101	0.077	0.103	+0.027
Few-NERD	7B base	0.313	0.409	0.412	0.415	+0.099
Few-NERD	14B base	0.401	0.459	0.451	0.460	+0.051
Few-NERD	OLMo-2 7B base	0.149	0.204	0.204	0.211	+0.055
Few-NERD	Phi-3.5-mini	0.351	0.348	0.348	0.357	-0.003
Few-NERD	Mistral-7B it	0.364	0.359	0.363	0.372	-0.001
Few-NERD	falcon3-3b-base	0.188	0.265	0.265	0.269	+0.077
Few-NERD	falcon3-3b-it	0.200	0.205	0.201	0.215	+0.001
Few-NERD	falcon3-7b-base	0.279	0.368	0.368	0.374	+0.089
Few-NERD	falcon3-7b-it	0.338	0.401	0.404	0.408	+0.066
Few-NERD	granite3.1-2b-base	0.148	0.286	0.283	0.287	+0.135
Few-NERD	granite3.1-2b-it	0.286	0.301	0.294	0.308	+0.008
Few-NERD	granite3.1-8b-base	0.297	0.360	0.362	0.371	+0.065
Few-NERD	granite3.1-8b-it	0.404	0.404	0.406	0.410	+0.003
Few-NERD	mistral-7b-base	0.237	0.351	0.355	0.361	+0.118
Few-NERD	olmo2-13b-base	0.256	0.337	0.333	0.339	+0.077
Few-NERD	qwen2.5-1.5b-base	0.123	0.170	0.165	0.178	+0.043
Few-NERD	qwen2.5-3b-base	0.198	0.284	0.281	0.288	+0.083
Few-NERD	smollm2-1.7b-base	0.156	0.242	0.240	0.246	+0.084
Few-NERD	smollm2-1.7b-it	0.207	0.247	0.253	0.256	+0.046
Few-NERD	smollm2-360m-base	0.051	0.112	0.055	0.118	+0.004
CoNLL04	0.5B	0.051	0.136	0.140	0.148	+0.090
GoEmotions	0.5B	0.120	0.115	0.102	0.122	-0.018
Inspec	0.5B	0.147	0.365	0.354	0.365	+0.208
WikiANN-ar	0.5B	0.099	0.225	0.239	0.247	+0.141
WikiANN-de	0.5B	0.389	0.451	0.458	0.460	+0.069
WikiANN-es	0.5B	0.418	0.450	0.459	0.466	+0.041
WikiANN-zh	0.5B	0.203	0.290	0.300	0.307	+0.097
CoNLL04	1.5B	0.153	0.202	0.193	0.214	+0.040
GoEmotions	1.5B	0.185	0.230	0.220	0.231	+0.035
Inspec	1.5B	0.398	0.433	0.431	0.436	+0.033
WikiANN-ar	1.5B	0.255	0.331	0.336	0.341	+0.081
WikiANN-de	1.5B	0.364	0.462	0.462	0.467	+0.098

Table 9 (continued)

CoNLL	0.5B base	0.364	0.356	0.291	0.367	-0.073
CoNLL	7B base	0.645	0.652	0.645	0.661	-0.000
CoNLL	14B base	0.656	0.720	0.711	0.720	+0.055
CoNLL	OLMo-2 7B base	0.462	0.492	0.498	0.508	+0.036
CoNLL	OLMo-2 7B it	0.508	0.566	0.562	0.577	+0.054
CoNLL	Phi-3.5-mini	0.452	0.515	0.498	0.515	+0.046
CoNLL	Mistral-7B it	0.463	0.487	0.483	0.490	+0.020
CoNLL	falcon3-3b-base	0.345	0.340	0.332	0.357	-0.013
CoNLL	falcon3-3b-it	0.333	0.348	0.344	0.350	+0.012
CoNLL	falcon3-7b-base	0.446	0.444	0.441	0.462	-0.005
CoNLL	falcon3-7b-it	0.588	0.595	0.605	0.608	+0.017
CoNLL	granite3.1-2b-base	0.334	0.328	0.326	0.341	-0.007
CoNLL	granite3.1-2b-it	0.539	0.561	0.548	0.568	+0.009
CoNLL	granite3.1-8b-base	0.544	0.532	0.542	0.550	-0.002
CoNLL	granite3.1-8b-it	0.673	0.674	0.675	0.679	+0.002
CoNLL	mistral-7b-base	0.465	0.459	0.459	0.470	-0.005
CoNLL	olmo2-13b-base	0.645	0.657	0.645	0.657	+0.000
CoNLL	qwen2.5-1.5b-base	0.488	0.497	0.495	0.502	+0.008
CoNLL	qwen2.5-3b-base	0.585	0.604	0.601	0.605	+0.017
CoNLL	smollm2-1.7b-base	0.252	0.243	0.244	0.253	-0.008
CoNLL	smollm2-1.7b-it	0.106	0.153	0.134	0.156	+0.028
CoNLL	smollm2-360m-base	0.178	0.188	0.186	0.194	+0.008
CoNLL	smollm2-360m-it	0.039	0.102	0.060	0.105	+0.021
Few-NERD	0.5B	0.039	0.111	0.076	0.113	+0.038
Few-NERD	1.5B	0.179	0.201	0.190	0.208	+0.011
Few-NERD	3B	0.204	0.255	0.248	0.261	+0.044
Few-NERD	7B	0.364	0.386	0.390	0.397	+0.026
Few-NERD	14B	0.327	0.390	0.393	0.394	+0.066
Few-NERD	0.5B base	0.050	0.101	0.077	0.103	+0.027
Few-NERD	7B base	0.313	0.409	0.412	0.415	+0.099
Few-NERD	14B base	0.401	0.459	0.451	0.460	+0.051
Few-NERD	OLMo-2 7B base	0.149	0.204	0.204	0.211	+0.055
Few-NERD	Phi-3.5-mini	0.351	0.348	0.348	0.357	-0.003
Few-NERD	Mistral-7B it	0.364	0.359	0.363	0.372	-0.001
Few-NERD	falcon3-3b-base	0.188	0.265	0.265	0.269	+0.077
Few-NERD	falcon3-3b-it	0.200	0.205	0.201	0.215	+0.001
Few-NERD	falcon3-7b-base	0.279	0.368	0.368	0.374	+0.089
Few-NERD	falcon3-7b-it	0.338	0.401	0.404	0.408	+0.066
Few-NERD	granite3.1-2b-base	0.148	0.286	0.283	0.287	+0.135
Few-NERD	granite3.1-2b-it	0.286	0.301	0.294	0.308	+0.008
Few-NERD	granite3.1-8b-base	0.297	0.360	0.362	0.371	+0.065
Few-NERD	granite3.1-8b-it	0.404	0.404	0.406	0.410	+0.003
Few-NERD	mistral-7b-base	0.237	0.351	0.355	0.361	+0.118
Few-NERD	olmo2-13b-base	0.256	0.337	0.333	0.339	+0.077
Few-NERD	qwen2.5-1.5b-base	0.123	0.170	0.165	0.178	+0.043
Few-NERD	qwen2.5-3b-base	0.198	0.284	0.281	0.288	+0.083
Few-NERD	smollm2-1.7b-base	0.156	0.242	0.240	0.246	+0.084
Few-NERD	smollm2-1.7b-it	0.207	0.247	0.253	0.256	+0.046
Few-NERD	smollm2-360m-base	0.051	0.112	0.055	0.118	+0.004
CoNLL04	0.5B	0.051	0.136	0.140	0.148	+0.090
GoEmotions	0.5B	0.120	0.115	0.102	0.122	-0.018
Inspec	0.5B	0.147	0.365	0.354	0.365	+0.208
WikiANN-ar	0.5B	0.099	0.225	0.239	0.247	+0.141
WikiANN-de	0.5B	0.389	0.451	0.458	0.460	+0.069
WikiANN-es	0.5B	0.418	0.450	0.459	0.466	+0.041
WikiANN-zh	0.5B	0.203	0.290	0.300	0.307	+0.097
CoNLL04	1.5B	0.153	0.202	0.193	0.214	+0.040
GoEmotions	1.5B	0.185	0.230	0.220	0.231	+0.035
Inspec	1.5B	0.398	0.433	0.431	0.436	+0.033
WikiANN-ar	1.5B	0.255	0.331	0.336	0.341	+0.081
WikiANN-de	1.5B	0.364	0.462	0.462	0.467	+0.098

Table 9 (continued)

WikiANN-de	0.5B	0.389	0.451	0.458	0.460	+0.069
WikiANN-es	0.5B	0.418	0.450	0.459	0.466	+0.041
WikiANN-zh	0.5B	0.203	0.290	0.300	0.307	+0.097
CoNLL04	1.5B	0.153	0.202	0.193	0.214	+0.040
GoEmotions	1.5B	0.185	0.230	0.220	0.231	+0.035
Inspec	1.5B	0.398	0.433	0.431	0.436	+0.033
WikiANN-ar	1.5B	0.255	0.331	0.336	0.341	+0.081
WikiANN-de	1.5B	0.364	0.462	0.462	0.467	+0.098

Table 10: Threshold stability in the sample budget.

Dataset	Model	θ^* measured at k					F_1 at k=60		
		5	10	20	40	60	θ=$\frac{1}{2}$	θ from k=10	oracle
CoNLL	1.5B	0.30	0.25	0.12	0.11	0.14	0.414	0.416	0.451
CoNLL	7B	0.50	0.55	0.68	0.66	0.71	0.636	0.639	0.646
CoNLL	14B	0.30	0.45	0.53	0.59	0.46	0.731	0.735	0.735
CoNLL	OLMo-2 7B it	0.30	0.25	0.23	0.29	0.28	0.431	0.547	0.552
Few-NERD	1.5B	0.30	0.25	0.28	0.14	0.17	0.164	0.194	0.212
Few-NERD	7B	0.10	0.35	0.12	0.14	0.09	0.355	0.368	0.391
Few-NERD	14B	0.10	0.15	0.12	0.04	0.04	0.319	0.356	0.367
Few-NERD	OLMo-2 7B it	0.10	0.05	0.03	0.01	0.01	0.008	0.156	0.194
CoNLL04	7B	0.30	0.25	0.23	0.14	0.11	0.210	0.224	0.229
GoEmotions	7B	0.90	0.65	0.82	0.89	0.97	0.318	0.317	0.328
Inspec	7B	0.30	0.45	0.42	0.39	0.38	0.436	0.439	0.443
WikiANN-de	7B	0.10	0.35	0.42	0.44	0.11	0.613	0.621	0.628
CoNLL04	14B	0.10	0.05	0.03	0.01	0.01	0.131	0.197	0.212
GoEmotions	14B	0.90	0.85	0.53	0.84	0.86	0.335	0.343	0.343
Inspec	14B	0.30	0.45	0.33	0.29	0.39	0.450	0.453	0.453
WikiANN-de	14B	0.10	0.25	0.23	0.21	0.23	0.693	0.705	0.711

Table 11: Precision, recall and F_1 at $k = 20$ relative to $k = 1$ under majority voting, with the union arm for contrast. Intervals are paired example bootstraps; significance is after Benjamini–Hochberg correction. Qwen2.5-Instruct shown; all families in Table F.5.

		k=1			majority k=20			union k=20		
Dataset	Model	P	R	F1	ΔP	ΔR	ΔF1	ΔP	ΔR	ΔF1
CoNLL	0.5B	0.183	0.337	0.237	+0.266	-0.021	+0.134	-0.130	+0.313	-0.140
CoNLL	1.5B	0.482	0.322	0.386	+0.195	-0.003	+0.048	-0.268	+0.322	-0.064
CoNLL	3B	0.548	0.532	0.540	+0.045	-0.012	+0.014	-0.195	+0.163	-0.071
CoNLL	7B	0.657	0.641	0.649	+0.034	+0.010	+0.021	-0.165	+0.123	-0.051
CoNLL	14B	0.797	0.697	0.744	+0.019	-0.009	+0.003	-0.139	+0.070	-0.035
Few-NERD	0.5B	0.074	0.047	0.058	+0.091	-0.027	-0.021	-0.038	+0.185	+0.005
Few-NERD	1.5B	0.180	0.138	0.156	+0.134	-0.013	+0.022	-0.097	+0.226	-0.021
Few-NERD	3B	0.291	0.169	0.214	+0.086	-0.030	-0.011	-0.119	+0.190	+0.019
Few-NERD	7B	0.419	0.332	0.370	+0.027	-0.024	-0.006	-0.116	+0.148	+0.001
Few-NERD	14B	0.482	0.258	0.336	+0.016	-0.022	-0.016	-0.083	+0.120	+0.052
WikiANN-ar	0.5B	0.210	0.145	0.171	+0.214	-0.087	-0.070	-0.117	+0.267	-0.019
WikiANN-de	0.5B	0.301	0.349	0.323	+0.184	-0.061	+0.038	-0.217	+0.381	-0.172
WikiANN-es	0.5B	0.412	0.282	0.335	+0.221	+0.063	+0.112	-0.204	+0.469	-0.009
WikiANN-zh	0.5B	0.227	0.225	0.226	+0.189	-0.076	-0.006	-0.120	+0.330	-0.046
WikiANN-ar	1.5B	0.270	0.276	0.273	+0.110	-0.074	-0.009	-0.137	+0.269	-0.059
WikiANN-de	1.5B	0.500	0.306	0.380	+0.126	-0.050	-0.016	-0.246	+0.305	-0.021
WikiANN-es	1.5B	0.454	0.271	0.340	+0.218	-0.033	+0.013	-0.190	+0.382	+0.036
WikiANN-zh	1.5B	0.318	0.198	0.243	+0.072	-0.046	-0.026	-0.105	+0.349	+0.062
WikiANN-ar	3B	0.424	0.432	0.428	+0.068	+0.011	+0.038	-0.162	+0.172	-0.063
WikiANN-de	3B	0.630	0.563	0.595	+0.029	+0.006	+0.016	-0.147	+0.113	-0.032
WikiANN-es	3B	0.686	0.647	0.666	+0.010	-0.012	-0.002	-0.159	+0.071	-0.058
WikiANN-zh	3B	0.494	0.565	0.527	+0.001	-0.004	-0.001	-0.171	+0.076	-0.098
WikiANN-ar	7B	0.477	0.446	0.461	+0.045	-0.002	+0.019	-0.183	+0.115	-0.075
WikiANN-de	7B	0.695	0.546	0.612	+0.021	+0.013	+0.016	-0.087	+0.090	+0.010
WikiANN-es	7B	0.744	0.682	0.711	+0.026	+0.002	+0.013	-0.106	+0.065	-0.023
WikiANN-zh	7B	0.490	0.450	0.469	+0.025	+0.004	+0.013	-0.108	+0.071	-0.028
WikiANN-ar	14B	0.643	0.497	0.560	+0.067	-0.002	+0.022	-0.140	+0.138	+0.001
WikiANN-de	14B	0.722	0.638	0.677	+0.017	-0.002	+0.006	-0.045	+0.077	+0.018
WikiANN-es	14B	0.779	0.576	0.662	+0.029	+0.018	+0.023	-0.069	+0.075	+0.018
WikiANN-zh	14B	0.523	0.508	0.515	+0.012	+0.011	+0.011	-0.076	+0.080	-0.008

Table 12: Calibration error, sharpness and the optimal threshold against sampling temperature, for base and instruction-tuned checkpoints.

quantity	arm	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$	$\tau = 1.0$	$\tau = 1.3$
calibration error	base	0.182	0.110	0.069	0.042	0.044
	tuned	0.286	0.221	0.173	0.123	0.089
mass at $f = 1$	base	0.156	0.076	0.037	0.009	0.001
	tuned	0.337	0.231	0.161	0.100	0.060
θ^*	base	0.394	0.343	0.307	0.229	0.131
	tuned	0.411	0.392	0.344	0.314	0.268

Table 13: The same effect per checkpoint pair. Calibration error falls as the sampling distribution is widened in every pair we ran, across six model families, and the optimal threshold falls with it.

checkpoint pair	ECE at $\tau = 0.3$	ECE at $\tau = 1.3$	ratio	θ^* shift
falcon3-3b	0.269	0.059	4.5×	0.308 → 0.167
falcon3-7b	0.216	0.039	5.6×	0.288 → 0.179
granite3.1-2b	0.254	0.067	3.8×	0.617 → 0.308
granite3.1-8b	0.255	0.062	4.1×	0.496 → 0.263
mistral-7b	0.302	0.094	3.2×	0.583 → 0.358
olmo2-13b	0.180	0.121	1.5×	0.375 → 0.108
olmo2-7b	0.191	0.059	3.2×	0.233 → 0.150
qwen2.5-0.5b	0.171	0.045	3.8×	0.367 → 0.117
qwen2.5-1.5b	0.204	0.047	4.3×	0.317 → 0.125
qwen2.5-14b	0.207	0.101	2.0×	0.325 → 0.183
qwen2.5-3b	0.238	0.065	3.7×	0.350 → 0.208
qwen2.5-7b	0.277	0.084	3.3×	0.400 → 0.250
smollm2-1.7b	0.218	0.046	4.8×	0.475 → 0.100
smollm2-360m	0.187	0.038	4.9×	0.508 → 0.075

Table 14: Modern instruction-tuned models: sharpness, optimal threshold, and the gain the optimal threshold buys over majority voting.

Model	Dataset	mass@f=1	θ^*	gain over $\frac{1}{2}$
Qwen3-8B	CoNLL	0.666	0.775	+0.006
Qwen3-8B	Few-NERD	0.352	0.075	+0.030
Qwen3-8B	GoEmotions	0.386	0.875	+0.013
Qwen3-32B	CoNLL	0.641	0.975	+0.009
Qwen3-32B	Few-NERD	0.347	0.175	+0.010
Qwen3-32B	GoEmotions	0.354	0.925	+0.009