Calibrated Typed Decisions: What a 7B Teacher's Soft Labels Do, and Don't, Teach a Small Classifier

Lagnajit Moharana · Technical report · November 2026
Dataset: huggingface.co/datasets/m-lagnajit/calibrated-typed-decisions
Code: github.com/Lagnajit09/100x_AI_ML/tree/main/calibrated-typed-decision
Models: m-lagnajit/ctd-student-f · m-lagnajit/ctd-student-e

Abstract

We study whether a small classifier can learn *honest confidence* from the probability distributions of a larger model. We build and publish *calibrated-typed-decisions*, a dataset of typed decisions (a text, a question and a set of options) drawn from six permissively licensed sources, in which every training question carries a soft label: the calibrated option probabilities of Qwen2.5-7B-Instruct, or human rater votes where they exist. We train ModernBERT-base students (about 150M parameters) on six targets, from one-hot human answers to blends with corrected teacher labels, and evaluate them three ways: on familiar data, on an unseen dataset of the same kind (SNIPS), and on a different kind of judgment (PubMedQA). Every comparison uses three seeds and three-seed ensembles, and a result counts as established only if it holds under both a seed-by-seed check and a paired bootstrap. We find that (i) on familiar data, hard labels with one post-hoc temperature are as good as any soft target; (ii) on a new label set, the corrected teacher distribution transfers best; (iii) on PubMedQA, teacher-shaped targets make students more honest but not more knowledgeable, and the best ensemble only just beats a label-frequency baseline (log loss 0.899 against 0.937); and (iv) a 2.6× larger student improves near transfer but not far transfer, while full fine-tuning makes the blended target worse on PubMedQA, an effect LoRA partly, but not reliably, avoids. We also report four rejected training sources, the checks that rejected them, and the errors caught along the way.

Contents

1. Introduction . . . 3
2. The dataset . . . 4
2.1 Typed decisions . . . 4
2.2 The teacher and its soft labels . . . 4
2.3 Version 1: five sources and a license screen . . . 5
2.4 Version 2: fixing the data . . . 5
2.5 Version 2.1: corrected labels . . . 6
3. Students and targets . . . 7
3.1 The student . . . 7
3.2 Six targets . . . 7
3.3 Training recipe . . . 8
4. Evaluation protocol . . . 9
4.1 Measures and baselines . . . 9
4.2 Dials . . . 9
4.3 Seeds, ensembles and the bootstrap . . . 9
4.4 The two-check rule, and pre-registration . . . 9
5. Results . . . 10
5.1 Version 1: the first answer was no . . . 10
5.2 Version 2, a single run . . . 10
5.3 One dial or twelve? . . . 10
5.4 Three seeds: what dissolved . . . 11
5.5 Ensembles: variance, not bias . . . 11
5.6 Run F . . . 12
5.7 Three kinds of new . . . 12
5.8 A larger student, fully fine-tuned and with LoRA . . . 14
6. Data selection, and four rejected sources . . . 15
7. Reproducibility, and errors caught . . . 16
8. Releasing two students . . . 17
8.1 Which students, and why . . . 17
8.2 What is in each repository . . . 17
8.3 Dials . . . 18
8.4 Checks before release . . . 18
8.5 Results of the published files . . . 18
9. Discussion . . . 19
9.1 Practical recommendations . . . 19
9.2 What the results mean . . . 19
10. Threats to validity . . . 20
10.1 Internal validity: are the differences caused by the targets? . . . 20
10.2 Construct validity: do the measures mean what we say? . . . 20
10.3 External validity: does it generalise? . . . 20
10.4 Statistical conclusion validity: are the differences real? . . . 20
11. Future work . . . 21
12. Conclusion . . . 21
Acknowledgements . . . 21
References . . . 22
Appendix A. Single students: mean (lowest–highest) over three seeds, 2 dials . . . 23
Appendix B. Ensembles of three seeds . . . 23
Appendix C. Kept epochs and compute . . . 24
Appendix D. Configuration . . . 24
Appendix E. Reproducing the results . . . 26
Appendix F. Artifacts . . . 26

1. Introduction

Many systems treat a model's confidence as a promise: a ticket router acts automatically above a threshold, a cascade asks an expensive model only when a cheap one is unsure, a moderation filter acts only when it is sure. That works only if the model is **calibrated**: among all the times it says 80%, it should be right about 80% of the time. Modern neural networks are often overconfident [Guo et al., 2017], and the standard repair is *temperature scaling*: after training, divide every score by one number fitted on held-out data. It is cheap, never changes which answer the model picks, and is a strong baseline.

Knowledge distillation [Hinton et al., 2015] suggests something richer. A teacher's full probability distribution says not only which answer is right but how plausible the alternatives are. Our hypothesis was that a small student trained on such **soft labels** from a well-calibrated teacher would learn calibrated uncertainty itself, and that this would show most on data unlike its training data, where a post-hoc temperature fitted on familiar data may not travel [Ovadia et al., 2019].

The first experiment rejected that hypothesis in its simple form, and most of this report is about what it took to find out what soft labels do and do not teach. Our contributions:

- **A published dataset** of typed decisions with teacher soft labels, human rater votes where available, and a corrected-label column; three tagged versions (v1.0, v2.0, v2.1), CC BY 4.0.
- **A controlled comparison of six training targets**, with model, data, steps and learning rates held fixed.
- **A three-way transfer evaluation:** familiar data, a new label set of the same kind (SNIPS), and a new kind of judgment (PubMedQA).
- **An evaluation protocol** designed against luck: a dial setup chosen by cross-fitting, three seeds, ensembles, a two-check rule for every claim, and questions written down before later experiments were run.
- **Capacity and adaptation experiments** with ModernBERT-large, fully fine-tuned and with LoRA.
- **Two released students**, one for each recommendation, ready to use in a few lines of code.
- **Negative results:** four candidate training sources rejected by pre-set checks, and a record of the errors found and corrected during the work.

The work happened in the order below; the report follows the same order, so each result can be read as the answer to the question that the previous one raised.

Table 1. *Project timeline.*

#	Stage	What was done	What came out of it
1	Dataset v1	five sources, teacher soft labels, license screen; published (v1.0)	the dataset format
2	Students v1	targets A, B, C; one run each	hard labels + one dial best; soft labels copy teacher mistakes
3	Dataset v2	rater votes, descriptions, hard distractors, SciCite, PubMedQA held-out (v2.0)	GoEmotions fixed; a far-transfer test
4	Students v2	targets A to E, one seed, 12 dials	D looked best at home; C and E transferred
5	Dataset v2.1	the corrected-label column behind target E	1,391 rows corrected
6	Step 1: dials	12 vs 2 vs no dials, by cross-fitting	2 dials, for every student
7	Step 2: seeds	seeds 1 and 2 for every target	D's win dissolved; luck up to 7.6 points far from home

#	Stage	What was done	What came out of it
8	Step 3: ensembles	three-seed averages and a paired bootstrap	fixes variance at home, not bias far away
9	Step 4: Run F	0.5 one-hot + 0.5 corrected, three seeds; later compared with C and E	best PubMedQA log loss; not proven better than D
10	Step 5: data search	WANLI, ContractNLI, COSMOS QA, HWU64 inspected against pre-set rules	all four rejected
11	Steps 6–7: SNIPS	a new label set, evaluation only; seeds 1–2 retrained with weights kept	no drop from seen data; E best
12	Step 8: ModernBERT-large	E and F fully fine-tuned, three seeds; E vs C and D added	helps near transfer, not far; F worse on PubMedQA
13	Step 9: LoRA	large F with LoRA, three seeds	partial recovery, not settled; high seed variance
14	Write-up and release	card updates, a blog post, this report, and two published students	

2. The dataset

2.1 Typed decisions

Every example is a **typed decision**: a text (the *state*), a question (the *instructions*) and a list of options. There are two types. **Choice** questions show 3 to 8 options in shuffled order. **Yes/no** questions ask whether one candidate label applies ("Is the customer's intent in this message: card arrival?") with options *false* and *true*; about 30% of questions are yes/no. Every source is converted into this one format, so a single model can answer all of them.

```
instructions: "Which banking intent best describes this customer message?"
state:        "I still haven't received my new card, what should I do?"
options:      card arrival · card delivery estimate · lost or stolen card · activate my card
gold:         card arrival
```

2.2 The teacher and its soft labels

The teacher is Qwen2.5-7B-Instruct [Qwen Team, 2024], run in 4-bit (NF4 quantization, 16-bit computation) on a free Colab T4, with the system prompt "*You are a careful text classifier. Reply with only the letter of the single best option.*" It reads each question **once**; instead of letting it generate an answer, we read the probabilities it assigns to each option's letter and renormalize them over the options shown. Questions where less than half of the teacher's probability fell on option letters are dropped, since it did not really answer. These raw chances are then adjusted by the teacher's own dials, fitted on validation questions separately for each source and question type: a temperature, plus a shift for yes/no questions (groups with fewer than 30 validation questions fall back to one pooled dial per type). The adjusted chances are stored as `teacher_probs`, and form the question's **soft label**.

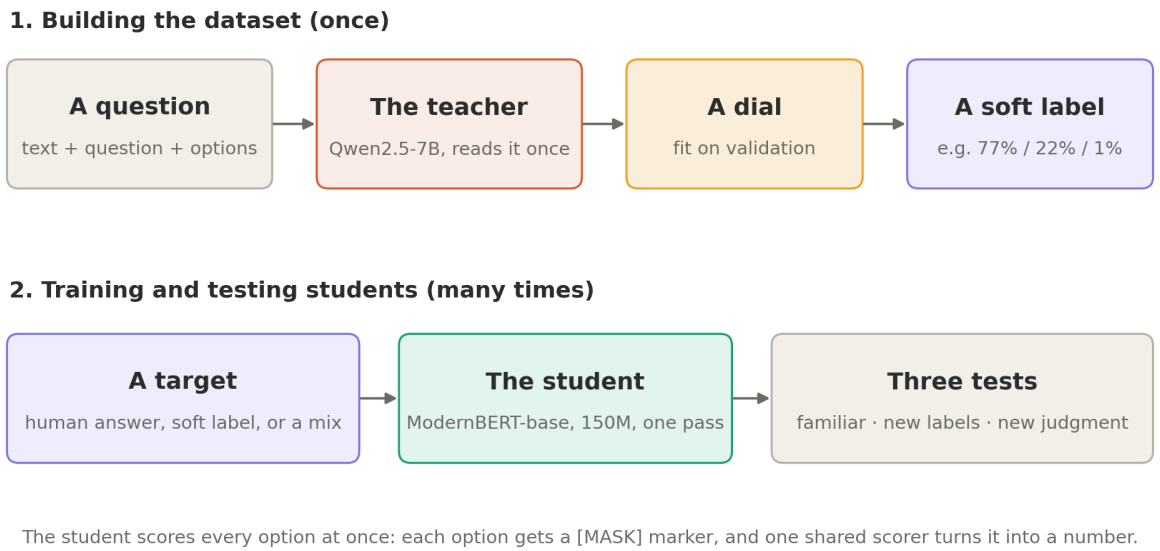


Figure 1. *The pipeline. The teacher scores each question once to build the dataset; students are then trained on different targets and tested three ways.*

2.3 Version 1: five sources and a license screen

Version 1 drew on five sources: CLINC150 [Larson et al., 2019], Banking77 [Casanueva et al., 2020], the English part of MASSIVE [FitzGerald et al., 2023], GoEmotions [Demszky et al., 2020] and Amazon Review Polarity [Zhang et al., 2015]. Every candidate source was screened against an allow-list of licenses (CC BY 4.0, 3.0 and 2.0, Apache 2.0, MIT, CC0), checked at the original source rather than at third-party copies; sources outside it were dropped. Examples rejected at this stage, or later while searching for a sixth source: AG News (license unknown), dair-ai/emotion ("other"), BoolQ and ARC (ShareAlike), SciQ (non-commercial) and OpenBookQA (license unclear). The v1 test split had 1,499 questions. Version 1 is tagged `v1.0`.

2.4 Version 2: fixing the data

The first student experiments (Section 5.1) showed that soft labels inherit the teacher's mistakes. Version 2 kept the student and training code fixed and changed the data:

- **Human votes for GoEmotions.** GoEmotions publishes each rater's individual labels. On 600 validation and test questions, the teacher's top pick disagreed with the official answer 38.5% of the time, while the raters' top pick matched it 96.7% of the time (average gap between teacher and rater distributions: 0.357). For GoEmotions, the soft label became the raters' vote shares, with 0.1 of a pseudo-vote added to every option so no option gets exactly zero.
- **Option descriptions**, so each option is shown as "label: description". Some sources' descriptions were written by hand; the rest were drafted by the teacher from real training examples and saved for review.
- **Hard distractors.** Half of the wrong options in each choice question are drawn from the 10 labels closest in meaning to the right one (by bge-small-en-v1.5 embedding similarity), the rest at random, so questions cannot be solved by easy elimination.
- **A sixth source,** SciCite [Cohan et al., 2019], and more questions: 3,000 training questions per source (2,000 for SciCite).

- **A held-out test from a different world:** the 1,000 expert-labelled questions of PubMedQA (PQA-L) [Jin et al., 2019]; its 211,000 automatically labelled questions were deliberately not used. No student is trained on it and no dial is fitted on it; the teacher's chances on it are stored without dials.

Training questions come from each source's training split; validation and test questions are disjoint halves of questions drawn from its evaluation split, with a random generator seeded per source. Version 2 is tagged `v2.0`. Splits: 16,995 training, 1,799 validation, 1,797 test and 1,000 held-out questions.

2.5 Version 2.1: corrected labels

On some teacher-labelled training questions, the teacher's top pick is wrong. Version 2.1 adds `corrected_soft_label`: on such rows, the values of the teacher's top pick and of the right answer are swapped, so the right answer receives the largest share while the rest of the teacher's distribution, including which wrong option it found most plausible, is kept. Rows where the right answer is already tied for the largest share are unchanged, and rows labelled by human raters are never changed. The correction changes 1,391 of 13,996 teacher-labelled training rows (9.9%), flagged by `soft_label_corrected`.

Table 2. *Sources. The first six provide training, validation and test questions; PubMedQA is the dataset's held-out split; SNIPS was used only for evaluation (Section 5.7) and is not distributed with the dataset.*

Source	Decision	License	Soft label from
CLINC150	intent of a request	CC BY 3.0	teacher
Banking77	banking intent of a message	CC BY 4.0	teacher
MASSIVE (English)	voice-assistant intent	CC BY 4.0	teacher
GoEmotions	emotion in a comment	Apache 2.0	human rater votes (v2 on)
Amazon Polarity	sentiment of a review	Apache 2.0	teacher
SciCite (v2 on)	why a paper is cited	Apache 2.0	teacher
PubMedQA, PQA-L (held-out)	does the evidence support yes / no / maybe?	MIT	evaluation only
SNIPS 2017 (not in the dataset)	voice-assistant intent, 7 new intents	CC0 1.0	evaluation only

On the v2 test split, the teacher scores accuracy 0.843, ECE 0.023 and log loss 0.448: well calibrated, but less accurate than any student (Section 5). On PubMedQA without dials it scores 0.647, 0.286 and 2.325: more accurate than the students, but badly overconfident.

3. Students and targets

3.1 The student

Students are ModernBERT-base [Warner et al., 2024] (about 150M parameters) with one added linear scorer. A question is laid out as a single sequence, with a `[MASK]` marker in front of each option:

```
[CLS] question [SEP] [MASK] option 1 [MASK] option 2 ... [MASK] option K [SEP] text [SEP]
question ≤ 96 tokens · each option ≤ 32 tokens · whole sequence ≤ 512 tokens
```

The body produces a vector for every token; the vectors at the K markers are gathered and scored by one shared linear layer (768 weights and one bias: 769 numbers). Because the scorer is shared, options can differ from question to question: the student judges how well each option fits rather than memorizing a label list.

Table 3. *Shapes in the student's forward pass, for a batch of B questions with up to K options.*

Tensor	Shape	Meaning
input_ids, attention_mask	(B, L)	questions padded to the longest in the batch
last_hidden_state	(B, L, H)	one vector per token (H = 768 base, 1,024 large)
marker_pos, option_mask	(B, K)	position of each option's marker; which of the K slots are real
markers	(B, K, H)	the vectors at the markers (torch.gather along the token axis)
scores	(B, K)	shared scorer applied to each marker; empty slots set to $-\infty$
chances	(B, K)	softmax over the options; empty slots get exactly zero

The loss is the cross-entropy between the student's chances p and a target distribution t over the shown options, averaged over the batch: $L = -\Sigma_k\ t_k \log p_k$. With a one-hot target this is $-\log$ p(right answer); with a soft target, every option contributes.

3.2 Six targets

Six runs differ **only** in their target. Let g be the one-hot human answer, s the soft label and c the corrected soft label:

Table 4. *The six training targets. On rows labelled by human raters, $s = c$.*

Run	Target	Idea
A	g	human answers only
B	0.9 g + 0.1 × uniform over the shown options	label smoothing [Szegedy et al., 2016]
C	s	the teacher's (or raters') soft label
D	0.5 g + 0.5 s	blend of A and C
E	c	the corrected soft label
F	0.5 g + 0.5 c	blend of A and E (added later, Section 5.6)

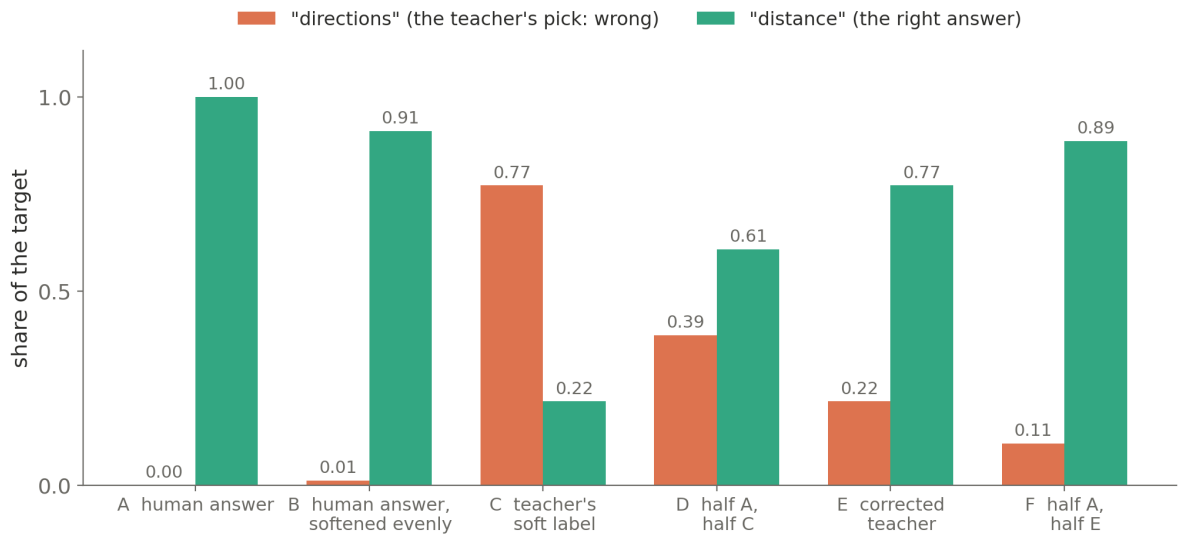


Figure 2. *The six targets on one real training question whose right answer is "distance" and whose teacher favoured "directions". C learns the teacher's mistake; D halves it; E and F move most of the weight to the right answer while keeping the teacher's runner-up.*

3.3 Training recipe

Table 5. *Training recipe, identical for every run unless stated.*

Setting	Value
Epochs, batch size	3 epochs, batch 16
Optimizer	AdamW, weight decay 0.01, 10% linear warmup, gradient clipping at 1.0, mixed precision
Learning rates	3e-5 for the pre-trained body, 1e-3 for the new scorer
Kept epoch	the epoch with the lowest validation log loss against the human answer
Seeds	0, 1, 2; a seed fixes the scorer's starting values, the batch order and the option shuffles, and is shared by all six targets
Hardware	Google Colab T4 (free tier); about 20 minutes per base run

4. Evaluation protocol

4.1 Measures and baselines

We report **accuracy**; **ECE**, the average gap between confidence and accuracy over 10 equal-width confidence bins, weighted by bin size; and **log loss**, the average of −log p(right answer), which rewards being right and honest at once and punishes confident mistakes most. ECE is unsigned: it does not say whether a model is over- or underconfident, so we read it together with reliability diagrams. Three tests are used: the 1,797 test questions (familiar data), 1,050 SNIPS questions (a new label set) and the 1,000 PubMedQA questions (a new kind of judgment). On PubMedQA, always answering "yes" scores accuracy 0.552, and always predicting the label frequencies scores log loss 0.937. On SNIPS, guessing evenly among the shown options scores 0.303 and 1.304.

4.2 Dials

After training, each student's scores are divided by a temperature T before the softmax; yes/no questions also get a shift on the "true" score. Dials are fitted on validation log loss and never change the top pick. Our first setup fitted one dial per source and question type (12 dials); Section 5.3 compares it with one dial per question type (2 dials) and with none, using **cross-fitting** on validation: fit on one half, score on the other, over 3 random splits and both halves. PubMedQA and SNIPS have no dials of their own: they use the pooled per-type dials fitted on validation.

4.3 Seeds, ensembles and the bootstrap

Every run is trained with three seeds. An **ensemble** averages the three dialed students' chances, option by option; this rule was fixed before any ensemble result was seen. Because −log is convex, the ensemble's log loss can never exceed its members' average (Jensen's inequality), which we check as a test of the code. Differences between ensembles are judged with a **paired bootstrap**: 1,000 resamples of the test questions, the same resample for both models; a difference is "better" or "worse" only if its 95% interval excludes zero.

4.4 The two-check rule, and pre-registration

A comparison is reported as **established** only if it holds under both checks: in **every seed** (comparing the two targets' students trained with the same seed, which share batch order and shuffles) and in the **ensemble bootstrap**. When the two models have different starting weights (base vs large), pairing seeds is arbitrary, so the seed check requires every seed of one to beat every seed of the other: all 9 pairings. From Run F onward, each experiment's question and pass/fail rule were written down before its results were seen, and the same was done for every candidate dataset inspected in Section 6.

5. Results

5.1 Version 1: the first answer was no

Table 6. *Version 1 results, single run. After its own dial, C reached an ECE of 0.027.*

v1 test (1,499 questions)	Accuracy	ECE	Log loss
Teacher (Qwen2.5-7B, 4-bit)	0.875	**0.019**	0.385
A: human answers	0.905	0.060	0.383
B: smoothed human answers	0.901	0.032	—
C: teacher soft labels	0.870	0.036	0.365
A + one dial	**0.907**	0.025	**0.297**

Soft labels made the student more honest than raw human answers (ECE 0.036 against 0.060), but one dial erased the difference: A with a dial was the most accurate model, had the best log loss, and beat its 50-times-larger teacher on both. The reason was clearest on GoEmotions (216 choice questions): the teacher scored 0.602, the soft-label student C 0.653, and the human-answer student A 0.810. In v1, the teacher's top pick was wrong on roughly 1 in 8 training questions, and C learned those mistakes with high weight. **A student cannot be much more right than the labels it learns from**: the value of soft labels depends on the teacher being right.

5.2 Version 2, a single run

Table 7. *Version 2, first run (seed 0), with 12 dials. The teacher's PubMedQA row is without dials.*

seed 0, 12 dials	test acc	test ECE	test LL	PubMedQA acc	PubMedQA ECE	PubMedQA LL
A	0.908	0.018	0.268	0.538	0.097	0.997
B	0.917	0.025	0.322	0.585	0.109	0.957
C	0.900	0.019	0.340	0.580	0.027	0.943
D	0.922	0.015	0.262	0.548	0.064	1.017
E	0.919	0.016	0.271	0.605	0.107	0.946
Teacher	0.843	0.023	0.448	0.647	0.286	2.325

The GoEmotions fix worked: C's accuracy on GoEmotions choice questions rose from 0.653 to 0.884 (on different test questions, so not a perfect comparison). Read naively, this single run suggested that the 50/50 blend D was the best target on familiar data (its accuracy beat A's in 992 of 1,000 bootstrap resamples, and its log loss beat B's in all 1,000) and that C and E transferred best (on PubMedQA, C beat A on accuracy, ECE and log loss, and E beat A on accuracy and log loss). The next three sections show which parts of that picture survived.

5.3 One dial or twelve?

Table 8. *Dial setups on the test split. Cross-fitted validation log loss, averaged over students: no dial 0.262, 12 dials 0.252, 2 dials 0.242.*

Test log loss (seed 0)	A	B	C	D	E
no dial	0.298	0.291	0.323	0.247	0.280
12 dials (per source and type)	0.268	0.322	0.340	0.262	0.271
2 dials (per question type)	**0.267**	**0.289**	**0.307**	**0.241**	**0.252**

Cross-fitting on validation preferred 2 dials for all five students, and the test split confirmed it for every student. Twelve dials overfit: with only a few hundred validation questions behind each, they fitted noise, and for B they were worse than no dial at all (cross-fitted 0.263 against 0.247). All later results use 2 dials.

5.4 Three seeds: what dissolved

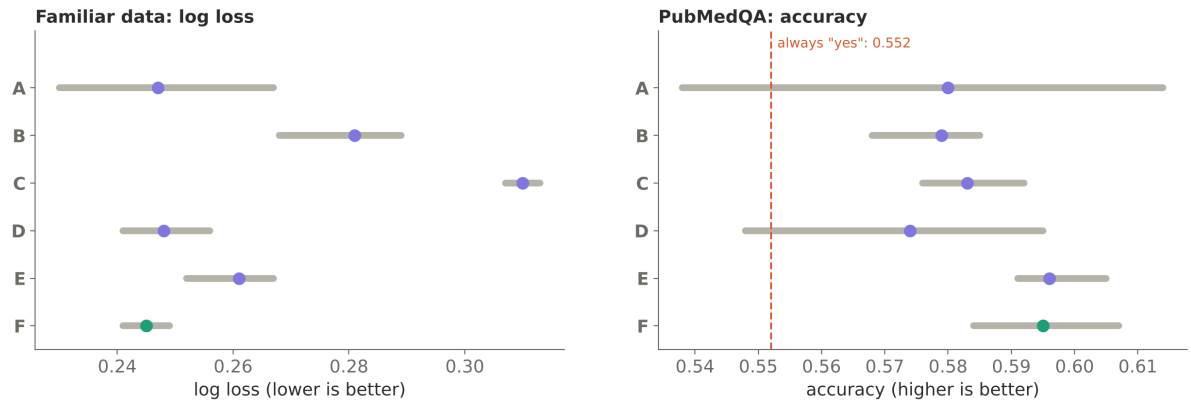


Figure 3. *Training luck. Dots: mean over three seeds; bars: lowest to highest seed. On familiar data the seeds agree closely; on PubMedQA, retraining alone moves A's accuracy from 0.538 to 0.614, and A's and D's ranges straddle the always-"yes" baseline.*

Retraining with new seeds moved familiar-data results by about one accuracy point, but held-out results by up to 7.6 points. Seed 0 had been D's luckiest seed and A's unluckiest, so D's single-run win dissolved: across seeds, D and A are tied on familiar data. What held in every seed: on familiar data, C is worse than A and B on accuracy and log loss; on PubMedQA, C is more honest than A (ECE and log loss), E has better log loss than A, and E beats B on accuracy and log loss. A's best epoch was 1, 2 and 2 across seeds, while B kept its last epoch in every seed.

5.5 Ensembles: variance, not bias

Table 9. *Ensembles for two runs (all runs in Appendix B). On familiar data every run's ensemble beat even its best seed; on PubMedQA only one run's did.*

Log loss	one student (mean)	best single	ensemble	ensemble, no dials
A, familiar	0.247	0.230	0.221	0.229
F, familiar	0.245	0.241	0.225	0.234
A, PubMedQA	0.970	0.945	0.964	0.924
F, PubMedQA	0.914	0.888	0.899	0.915

On familiar data, averaging three seeds improved every run beyond its best single student, and added 0.3 to 1.0 accuracy points. On PubMedQA it barely helped. Averaging cancels disagreement between students (variance); it cannot cancel a mistake they all share (bias), and on PubMedQA all students lack the same skill (figure below). The ensemble bootstrap reproduced the seed-by-seed pattern of Section 5.4 by a different route: hard targets lead on familiar data, and C, D and E are more honest than A on PubMedQA. One side effect: averaging makes chances less extreme, so an ensemble of students that are already honest drifts slightly underconfident. A's ensemble ECE rose from 0.011 (single-student mean) to 0.018, while every other run's fell.

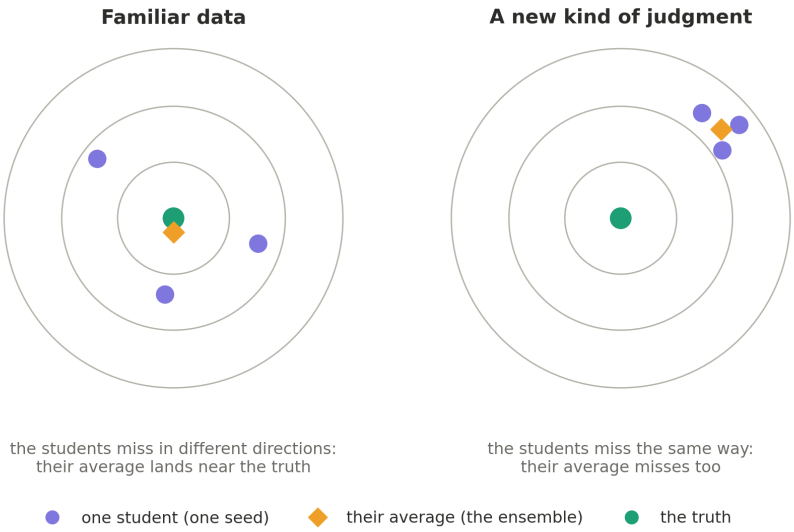


Figure 4. *Why averaging helps on familiar data but not on a new kind of judgment: independent misses cancel, shared misses do not.*

5.6 Run F

D's one known weakness is that on the 1 in 10 rows where the teacher was wrong, it still gives the wrong answer about 39% of the target. Run F blends the human answer with the corrected label instead (Table 3). The question, written in advance, was whether F beats D under both checks. It did not: the average gap favoured F in five of six comparisons, but none held in every seed and in the ensemble bootstrap. F did have the best familiar-data accuracy (ensemble 0.928) and ECE (0.006), the smallest seed-to-seed spread in accuracy and ECE, and the best PubMedQA log loss of any run (ensemble 0.899). The comparisons of F with E and C were omitted at first and added afterwards; F was not reliably worse than any target on familiar data or PubMedQA.

5.7 Three kinds of new

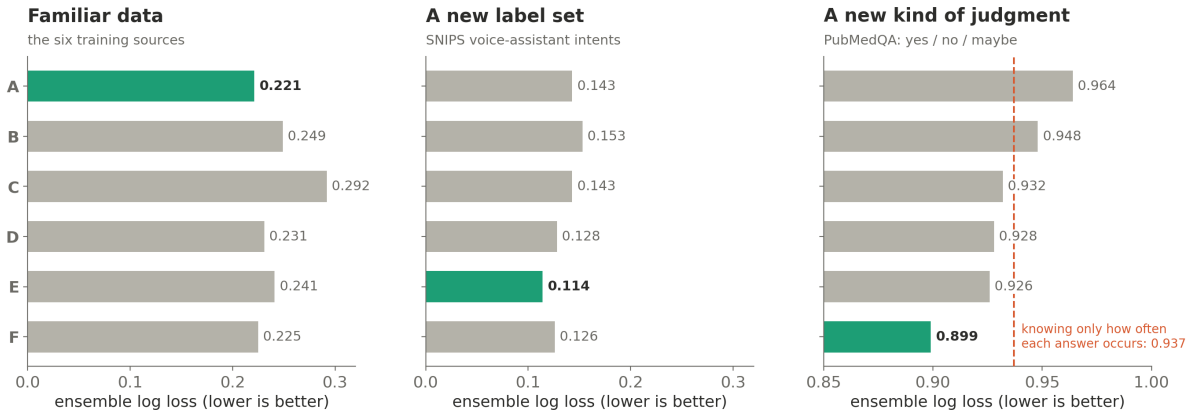


Figure 5. *Ensemble log loss of every run on the three tests. Best per test in green. On PubMedQA (axis from 0.85), the dashed line is the label-frequency baseline.*

Familiar data. A with dials and F are statistically tied (ensemble log loss 0.221 and 0.225); the raw soft label C is reliably the weakest. Every student beats the teacher, scoring 0.89 to 0.92 against its 0.843.

A new label set. SNIPS [Coucke et al., 2018] has 7 voice-assistant intents the students never saw; none of the 1,050 requests appear word-for-word in the training data. We took 150 requests per intent and built questions in the dataset's own format: descriptions we wrote for each intent, 3 to 7 options, and about 30% yes/no (734 choice and 316 yes/no questions), with a fixed seed so that every student answers identical questions. Before answering, each seed-0 student was reloaded from its saved weights and had to reproduce its saved test predictions (largest gap: 0.0000). Accuracy barely changes from MASSIVE, a seen dataset of the same kind, to SNIPS:

Table 10. *Near transfer. SNIPS's seven intents are easier to tell apart than MASSIVE's close-label distractors, so part of "no drop" reflects easier questions.*

Accuracy (single students, 3 seeds)	A	B	C	D	E	F
MASSIVE (seen)	0.948	0.951	0.921	0.955	0.949	0.954
SNIPS (never seen)	0.949	0.948	0.951	0.953	0.960	0.950

On SNIPS, **E** has the best ensemble (accuracy 0.965, log loss 0.114). Under both checks it beat A (ECE, log loss), B (accuracy, ECE, log loss), C (log loss), D (accuracy) and F (log loss), and nothing beat it. Mixing in the one-hot answer helps at home and hurts here: F is worse than E on SNIPS in every seed on all three measures.

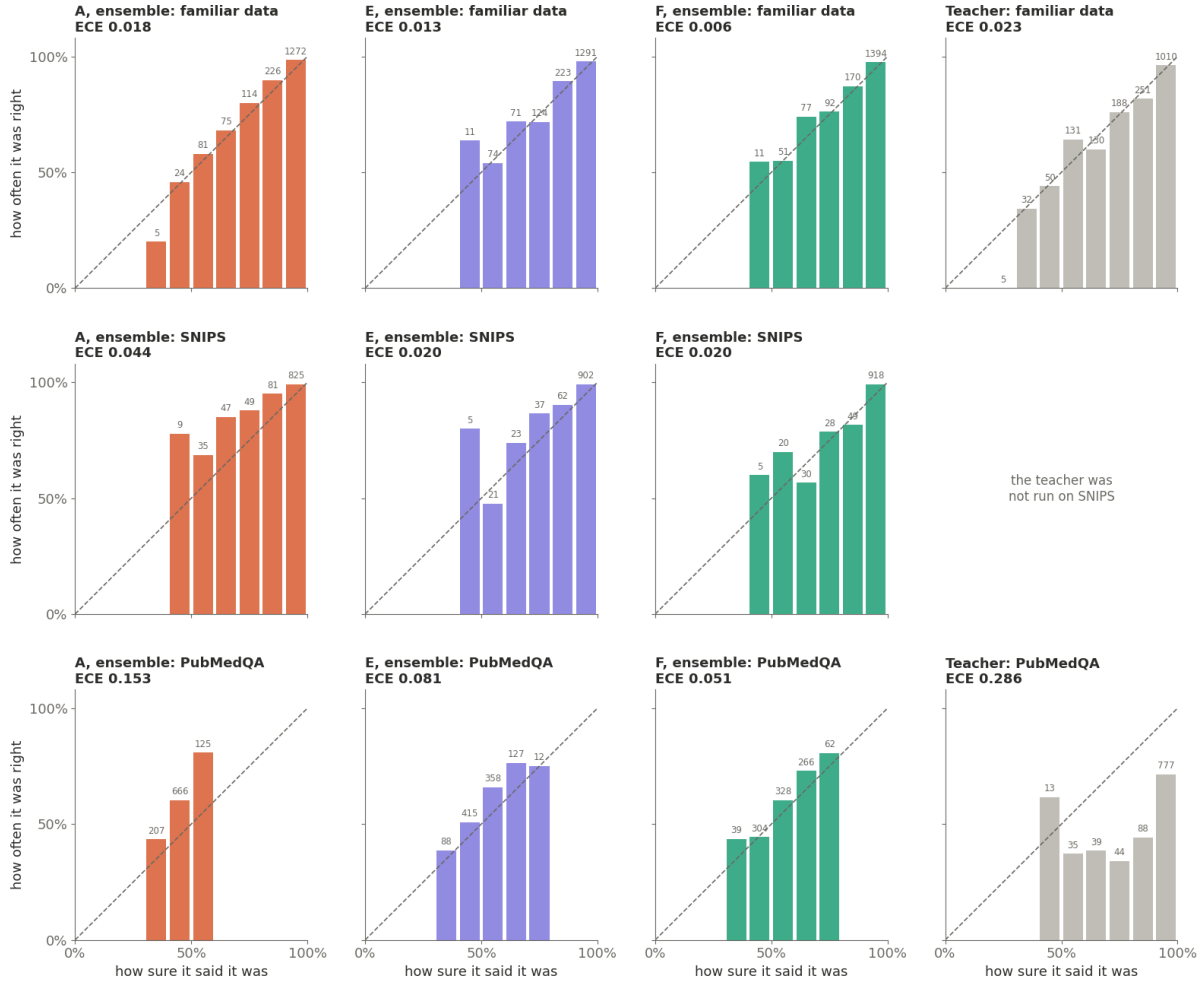


Figure 6. *Reliability diagrams for the ensembles of A, E and F and for the teacher, on the three tests (the teacher was not run on SNIPS). Bar labels: number of questions per confidence bin.*

A new kind of judgment. On PubMedQA, F's ensemble is the first to clear the label-frequency baseline clearly (log loss 0.899 against 0.937; accuracy 0.595 against 0.552). The reliability diagrams show what did transfer. A is badly miscalibrated (ECE 0.153), but

underconfident: it never says more than 60%, yet in its 50 to 60% bin it is right 81% of the time. Its dials, fitted on familiar data where hard-label students need strong toning down, flatten it too much here: without dials, A's ensemble log loss would be 0.924 rather than 0.964. The same effect appears, milder, on SNIPS (A's ECE 0.044 against 0.020 for E and F). F stays close to the diagonal (ECE 0.051), and C, D and F are reliably more honest than A on PubMedQA. The teacher is the overconfident one: 777 of its 1,000 PubMedQA answers come with at least 90% confidence, and only about 71% of those are right.

5.8 A larger student, fully fine-tuned and with LoRA

We repeated E and F with ModernBERT-large (about 395M parameters, 28 layers, H = 1,024), same recipe and data, three seeds each. Batches of 16 were processed as two pieces of 8 with gradient accumulation, which gives exactly the same training step (the loss is a batch mean, so weighting each piece by its share recovers it). The question, written before training: *does the large student beat the base student with the same target on PubMedQA log loss, under both checks?* Then, because fully fine-tuned large F got worse on PubMedQA, we trained F once more with LoRA [Hu et al., 2022]: rank 16, alpha 32, dropout 0.05, on the attention (Wqkv, Wo) and feed-forward (Wi, Wo) layers of all 28 blocks, 7,196,672 trainable numbers (1.79% of the body), adapter learning rate 2e-4. Its question: *does LoRA-F beat fully fine-tuned large F on PubMedQA log loss, under both checks?* Before its runs, the LoRA code checked that the named layers existed, and after training that the B matrices, which start at exactly zero, had moved.

Full fine-tuning of the large model peaked at 9.3 GB of GPU memory in a short trial and took about 0.6 to 0.7 seconds per training step, roughly 40 to 45 minutes per run. The large runs kept their final epoch in four of six cases (base runs: eight of eighteen), so they were still improving on validation when training stopped. LoRA ran at about the same speed (0.58 seconds per step); processing a batch as one piece of 16 ran out of memory on long questions, and two pieces of 8 fixed it (Section 7). LoRA's runs kept epochs 3, 2 and 2.

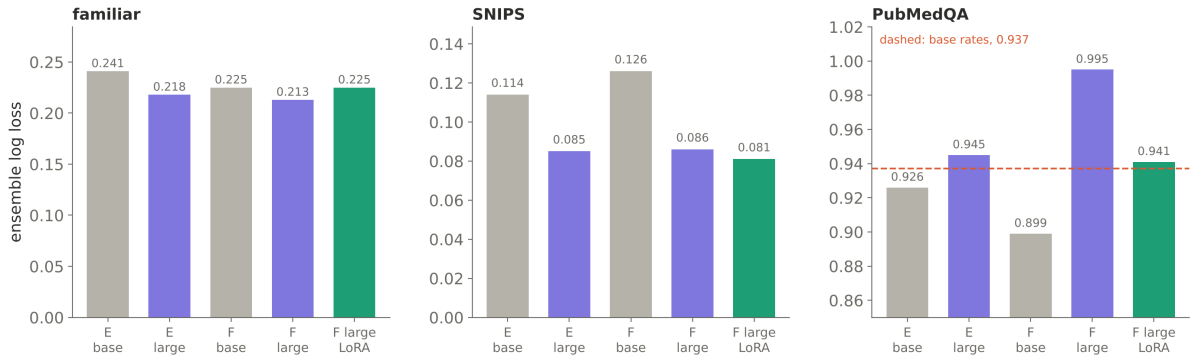


Figure 7. *Ensemble log loss for base and large students. The PubMedQA axis starts at 0.85; the dashed line is the label-frequency baseline.*

Table 11. *Larger students. Under both checks, large beat base on SNIPS log loss for E and F, on SNIPS accuracy for F, and on familiar log loss for E; LoRA-F beat base F on SNIPS accuracy and log loss.*

Ensembles	familiar acc	familiar LL	SNIPS acc	SNIPS LL	PubMedQA acc	PubMedQA LL
E, base	0.918	0.241	0.965	0.114	0.587	0.926
E, large	0.922	0.218	0.972	0.085	0.571	0.945
F, base	0.928	0.225	0.958	0.126	0.595	0.899
F, large	0.929	0.213	0.976	0.086	0.561	0.995
F, large, LoRA	0.918	0.225	0.972	0.081	0.594	0.941

Both pre-registered answers are no. No larger student was reliably better than its base counterpart on PubMedQA, and fully fine-tuned large F was clearly worse (log loss 0.995 against 0.899, worse in 8 of 9 seed pairings and in every bootstrap resample; ECE worse in all 9 pairings). LoRA recovered much of the loss: its ensemble beat full fine-tuning on PubMedQA accuracy, ECE and log loss in 1,000 of 1,000 resamples. But its three seeds ranged from 0.866 to 1.103 in log loss (the best and the worst PubMedQA students we trained), so only 5 of 9 pairings agreed, and the seed check did not settle it. LoRA also leaned towards learning less at home (ensemble accuracy 0.918 against 0.929), again without settling. On PubMedQA, the small base F remains the best student.

6. Data selection, and four rejected sources

The PubMedQA gap is a missing skill: none of the six training sources asks whether a text supports a claim. We therefore searched for a training source of that shape, not biomedical (so that a PubMedQA gain would show the *skill* transferring), and inspected each candidate against rules fixed before inspection.

Table 12. *Candidate sources. Each inspection took minutes and no GPU; building any rejected source into the dataset would have cost hours of teacher scoring and retraining.*

Candidate	Outcome	Measurement
WANLI [Liu et al., 2022]	rejected: no license stated	license field blank on every copy; none on the project pages
ContractNLI [Koreeda and Manning, 2021]	rejected: 2 of 4 rules failed (CC BY 4.0 confirmed)	statement-only baseline 0.681 vs 0.500 majority (dev) and 0.659 vs 0.463 (test), against a limit of +10 points; at 1,600 characters, 70.5–81.7% of evidence rows kept their evidence, against 80%; 17 fixed hypotheses, 5 never contradicted
COSMOS QA [Huang et al., 2019]	rejected: 1 of 8 rules failed (CC BY 4.0)	"None of the above" in 75.5% of training questions, right in 7.8% of them (rule: 10–40%); a word-overlap heuristic scored 0.285 on training and 0.164 on validation, so the splits differ
HWU64	rejected as an unseen test	the text release of SLURP, from which MASSIVE (a training source) was built
SNIPS 2017	accepted, evaluation only	CC0 1.0 on the authors' repository; 0 of 1,050 requests found word-for-word in training data

The two NLI candidates illustrate the annotation-artifact problem [Gururangan et al., 2018]: when a label can be guessed from part of the input, a student learns the cue rather than the judgment. The checklist distilled from these inspections, in order of cost: license at the source (blank counts as no; ShareAlike and non-commercial fail; copies can be wrong), provenance and label quality, fit to the intended skill, shortcut baselines, overlap with existing sources and test sets, and enough rows after filtering.

7. Reproducibility, and errors caught

Retrained seeds. Model weights had been kept only for seed 0, so evaluating seeds 1 and 2 on SNIPS required retraining them with the same recipe and seeds. GPU arithmetic is not fully deterministic, so the reruns are fresh samples of the same recipe: on the test split they agreed with the originals on 91.8% to 98.4% of top picks, and 8 of 12 fell within the original three-seed range. The latter is a weak yardstick: a fresh draw lands outside the range of n earlier draws with probability 2/(n + 1), which is 50% for n = 3.

Errors found and fixed during the work:

- The SNIPS question generator was first seeded with the *training* seed, which would have given each seed's students different questions. It now has its own fixed seed and prints a fingerprint of the question set.
- Processing LoRA batches as one piece of 16 ran out of GPU memory: LoRA saves optimizer memory, but activations double with the piece size and LoRA keeps extra dropout copies. Pieces of 8 give the identical step.
- An early reading called A "overconfident" on PubMedQA from its ECE alone; the reliability diagram showed the opposite. ECE has no sign.
- A recommendation that F was "never reliably worse than any target" was drafted before F had been compared with C and E; the comparisons were added, and the claim restated to what was tested.
- A test of the analysis code passed or failed at random because its fake data was seeded with Python's randomized `hash()`; it now uses a stable checksum.

The dataset card is generated, not written. Every claim in its results section is computed from the results files, and the generating notebook refuses to publish if a claim stops holding: for example, if any target becomes reliably better than the recommended one, or if two results files disagree about the same students. The card's results section was regenerated four times as the work progressed:

- after Steps 2 and 3: three-seed and ensemble results, and an automatically built list of what held up under both checks;
- after Run F: F recommended, with the targets it was compared with and its steadiest measures computed from the files;
- after SNIPS: a third kind of test and a conditional recommendation (F for familiar data, E for new label sets);
- after Steps 8 and 9: the larger students, and E compared with every other target.

8. Releasing two students

We published the two recommended base students as m-lagnajit/ctd-student-f and m-lagnajit/ctd-student-e, so that the results can be used, not only read. This section records what was released, how it was packaged, and how we checked that the published files behave exactly like the students analysed in this report.

8.1 Which students, and why

- **F (base)**, for data like the six training sources: the best familiar-data accuracy (ensemble 0.928) and ECE (0.006), never reliably beaten on familiar data, and the best PubMedQA log loss of any target (0.899).
- **E (base)**, for new label sets of the same kind: the best SNIPS ensemble (log loss 0.114), reliably ahead of every other target on at least one measure, and never reliably behind.
- **Not the large students.** They improved near transfer but not far transfer, fully fine-tuned large F was worse on PubMedQA, LoRA's results were too seed-dependent to recommend, and each costs about 2.6 times as much to run.
- **Three seeds each, not one.** Section 5.4 showed how much a single seed can mislead; each repository holds three students and averages their dialed chances, as in this report's ensembles.

8.2 What is in each repository

Table 13. *The files in each model repository.*

File	Contents
seed0.safetensors, seed1.safetensors, seed2.safetensors	the three students' weights, in 16-bit, about 300 MB each. Seed 0 is the original student; seeds 1 and 2 are the retrained students of Section 7, the only ones whose weights were kept
dials.json	each seed's dials: a temperature for choice questions, a temperature and a "yes" shift for yes/no questions
config.json	the base model and the token limits (512 per sequence, 96 for the question, 32 per option)
ctd_student.py	a small package that builds the input exactly as in training, applies each seed's dial and averages the seeds
README.md	the model card: intended use, usage, results measured on these files, limitations, license and citation

Why 16-bit. The original seed-0 students were saved in 32-bit (about 600 MB each) and the retrained ones in 16-bit (about 300 MB). Storing all three in 16-bit brings a repository from about 1.2 GB to about 0.9 GB. The package loads the weights back into 32-bit and computes exactly as in training, so only storage is rounded: 16-bit keeps about three significant digits per number, and Section 8.4 shows that no answer changed.

Why safetensors. PyTorch's usual `.pt` files are written with Python's pickle, which can run code when a file is loaded. A safetensors file holds only a small header (names, shapes, types) and the raw numbers, so loading it can never execute anything; it is the standard format on Hugging Face.

8.3 Dials

The experiments never stored their dials: every analysis refitted them from saved validation predictions. For the release, each published student was run on the 1,799 validation questions and its dials were fitted with exactly the code behind this report's 2-dial results, so that they match the published 16-bit weights.

Table 14. *The published dials.*

Student	choice temperature (seeds 0 / 1 / 2)	yes/no temperature, shift (seeds 0 / 1 / 2)
F	0.786 / 0.821 / 0.803	0.821, +0.2 / 0.803, +0.9 / 0.858, +0.5
E	0.644 / 0.704 / 0.704	0.630, +0.65 / 0.673, +0.9 / 0.735, +0.55

Every temperature is below 1: on their own, these students are slightly *underconfident*, and their dials sharpen them. E needs more sharpening than F, since its target has no one-hot half and its raw chances are softer, more like the teacher's. Every yes/no shift is positive: the students say "yes" slightly too rarely on their own. The three seeds' dials are close to each other, a sign that they capture something stable about each target.

8.4 Checks before release

Table 15. *Checks run before publishing. The first two ran without a GPU; the rest on the real students.*

Check	What it tests	Result
Same input	the package's token sequence and marker positions against the training notebook's build_input, for 300 questions of both types, with and without descriptions	identical for 300 of 300, including 187 cut to fit 512 tokens
Same dials and measures	applying and fitting dials, and accuracy, ECE and log loss, against the training notebook's code	identical
Complete students	every weights file contains both the body and the scorer	all six
Same SNIPS questions	a fingerprint of the rebuilt question set against the experiments'	b0096e8b4312, identical
Reproduction	each student, loaded through the package, against its saved predictions on 64 test questions: largest score gap below 0.05 and the same top pick on all 64	gaps 0.009–0.016; 64 of 64 for all six
Ensemble arithmetic	each ensemble's log loss is no worse than its members' average (Jensen)	passed on all three tests
A stranger's first run	the model card's usage example in a brand-new Colab session	worked; see below

The reproduction gaps of about 0.01 are what 16-bit rounding should produce. In the fresh-session test, student F gave "card arrival" 98.0% for a customer who had not received a new card, and 96.0% "yes" when asked whether that was the intent. That test also exposed one flaw in the model card: its usage example made two calls in one notebook cell, and a notebook displays only the last result; both calls now print.

8.5 Results of the published files

Table 16. *Measured through the package's public predict() function, on the files people download.*

Ensemble	accuracy	ECE	log loss	this report's log loss
F, familiar data	0.929	0.015	0.223	0.225
F, SNIPS	0.958	0.020	0.126	0.126
F, PubMedQA	0.608	0.085	0.910	0.899
E, familiar data	0.920	0.014	0.237	0.241
E, SNIPS	0.965	0.020	0.114	0.114
E, PubMedQA	0.607	0.097	0.918	0.926

On SNIPS, both published ensembles match this report to three decimals: they are the same students, and after 16-bit rounding, the package's own input building, newly fitted dials and a different code path, they give the same results. On familiar data and PubMedQA they differ slightly, because this report's ensembles there used the original seeds 1 and 2. The differences are largest on PubMedQA, where training luck is largest (Section 5.4): F's published log loss of 0.910 lies inside the range its single students covered in this report (0.888 to 0.953). The recommendations are unchanged: F has the better familiar-data accuracy and PubMedQA log loss, E the better SNIPS results. The model cards report these measured numbers.

9. Discussion

9.1 Practical recommendations

Table 17. *Recommendations, as published on the dataset card. In every case: 2 dials, and an ensemble of three seeds if affordable.*

If the student will be used on...	Use	Evidence
data like the training sources	F (A with a dial is tied)	nothing reliably better than F on familiar data
a new label set of the same kind	E	best SNIPS ensemble; beat every other target under both checks
a new kind of judgment	base F, but expect little	best PubMedQA log loss (0.899), barely below base rates (0.937)
any of these, if a larger model is affordable	large E or F for near transfer only	better on SNIPS under both checks; not better on PubMedQA

The two recommended base students are published, with three seeds each: see Section 8.

9.2 What the results mean

Calibration is not knowledge. In Murphy's [1973] decomposition, a proper score splits into reliability (calibration) and resolution (how much predictions separate outcomes). Dials and soft targets improved reliability; on PubMedQA, nothing we tried improved resolution much. A forecaster who always predicts the base rate is perfectly calibrated and uninformative, and the PubMedQA students sit near that corner.

Sharpness trades home performance for transfer. The more one-hot weight a target carries, the better it does on familiar data and the worse on a new label set (A and F lead at home, E on SNIPS). A larger, fully fine-tuned model amplifies the trade-off for the sharper target. This fits the view that fine-tuning distorts pre-trained features in ways that hurt out-of-distribution performance [Kumar et al., 2022], and LoRA's partial recovery fits "learns less, forgets less" [Biderman et al., 2024], though our evidence for the latter is not settled.

Luck is large exactly where it matters. Seed-to-seed variation was small on familiar data and several accuracy points on PubMedQA [cf. Dodge et al., 2020]; a single run produced a confident wrong conclusion (D best) that three seeds removed. Ensembles reduce such variance [Breiman, 1996; Lakshminarayanan et al., 2017] but not shared bias. Capacity helped near transfer, not far transfer [cf. Hendrycks et al., 2020]: the remaining lever is training data that teaches the missing judgment.

10. Threats to validity

10.1 Internal validity: are the differences caused by the targets?

- Everything except the target is fixed, and seeds are shared across targets. But the same validation set chooses each run's kept epoch and fits its dials. This cannot leak test information, but it can favour targets whose validation behaviour happens to generalise.
- LoRA differs from full fine-tuning in its learning rate as well as in which weights train; the two are inherent to the method and cannot be separated in our design.
- SNIPS results for seeds 1 and 2 come from retrained students. GPU arithmetic is not fully deterministic, so they are new samples of the same recipe rather than the students evaluated on test and PubMedQA.
- Some comparisons (F against C and E; E against C and D) were added after other results were seen. All were reported whatever their outcome, and claims were restricted to the comparisons actually run.

10.2 Construct validity: do the measures mean what we say?

- ECE depends on its binning (10 equal-width bins) and has no sign; log loss mixes skill with honesty. We read both together with reliability diagrams, and never ECE alone.
- "Honest" means calibrated on these measures. Choice and yes/no questions are pooled in every metric, although they have different base rates and different dials.
- PubMedQA's "maybe" is ambiguous even for experts, which caps any model's accuracy; SNIPS's seven intents are easier to tell apart than the training sources' close-label distractors.

10.3 External validity: does it generalise?

- One teacher, one student family, English only, closed-set text decisions, and one far-transfer dataset. Results may differ for other teachers, generative students or other domains.
- The pre-training data of ModernBERT and Qwen may include some of these public sources, PubMed abstracts among them. This could raise absolute scores, though it affects every target alike.

10.4 Statistical conclusion validity: are the differences real?

- Three seeds per run; seed-to-seed variation is large far from home (Section 5.4).
- Many comparisons are made (up to 39 per test) without a formal multiple-comparison correction. The two-check rule reduces false positives but does not eliminate them.
- The bootstrap treats questions as independent, although some questions share similar texts.
- The all-pairings seed check is conservative for high-variance methods such as our LoRA runs: results it leaves "not settled" may still be real.

11. Future work

- A training source that teaches evidence judgment (WANLI, if its license can be resolved), retraining A, E and F.
- A sweep of the one-hot share between E (0%) and F (50%), and beyond, to map the trade-off.
- Leave-one-source-out evaluation for near transfer without easier questions.
- Dials that travel: detecting distribution shift, or shrinking temperatures towards 1 on unfamiliar data.
- A pre-registered statistical test suited to high-variance methods, with more seeds for LoRA.

12. Conclusion

We asked whether a small classifier trained on a larger model's soft labels learns honest confidence. On data like its training data, it does not need to: human answers with one post-hoc temperature are as good as any soft target, and the raw soft label is worse, because it carries the teacher's mistakes. Soft labels earn their place away from home. On a new label set of the same kind, the corrected teacher distribution (E) is the best target we tested, and on a new kind of judgment, every teacher-shaped target makes students more honest than human answers alone do.

What soft labels cannot do is supply knowledge the training data never asks for. On PubMedQA the best students are well calibrated and barely more informative than the label frequencies; ensembles, dials, better targets and a 2.6× larger student did not change that, and full fine-tuning of the larger student made the blended target worse. The remaining lever is data that teaches the missing judgment.

Methodologically, the most important result may be the least specific one: a single training run produced a confident conclusion that three seeds overturned, and seed-to-seed variation far from home was larger than most of the differences we set out to measure. Every claim in this report holds in every seed and in a bootstrap, or is labelled as unsettled. That discipline, with questions written down before results and negative results reported alongside positive ones, is what makes the positive results believable.

Acknowledgements

The experiments in this report were designed, implemented, analysed and written up with the assistance of Claude (Anthropic), which helped write the code and its tests, interpret the results and draft the text. All runs were executed and checked by the author on Google Colab.

References

Anthropic (2026). Claude Opus 5.5 [large language model]. Assisted with experiment design, code, analysis and the writing of this report. anthropic.com/claude.

Biderman, D. et al. (2024). LoRA learns less and forgets less. *Transactions on Machine Learning Research*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2).

Casanueva, I. et al. (2020). Efficient intent detection with dual sentence encoders. *NLP4ConvAI Workshop*.

Cohan, A. et al. (2019). Structural scaffolds for citation intent classification in scientific publications. *NAACL*.

Coucke, A. et al. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv:1805.10190*. Data: github.com/sonos/nlu-benchmark.

Demszky, D. et al. (2020). GoEmotions: a dataset of fine-grained emotions. *ACL*.

Dodge, J. et al. (2020). Fine-tuning pretrained language models: weight initializations, data orders, and early stopping. *arXiv:2002.06305*.

FitzGerald, J. et al. (2023). MASSIVE: a 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *ACL*.

Guo, C., Pleiss, G., Sun, Y. and Weinberger, K. Q. (2017). On calibration of modern neural networks. *ICML*.

Gururangan, S. et al. (2018). Annotation artifacts in natural language inference data. *NAACL*.

Hendrycks, D. et al. (2020). Pretrained transformers improve out-of-distribution robustness. *ACL*.

Hinton, G., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv:1503.02531*.

Hu, E. J. et al. (2022). LoRA: low-rank adaptation of large language models. *ICLR*.

Huang, L. et al. (2019). Cosmos QA: machine reading comprehension with contextual commonsense reasoning. *EMNLP*.

Jin, Q. et al. (2019). PubMedQA: a dataset for biomedical research question answering. *EMNLP*.

Koreeda, Y. and Manning, C. D. (2021). ContractNLI: a dataset for document-level natural language inference for contracts. *Findings of EMNLP*.

Kumar, A. et al. (2022). Fine-tuning can distort pretrained features and underperform out-of-distribution. *ICLR*.

Lakshminarayanan, B., Pritzel, A. and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*.

Larson, S. et al. (2019). An evaluation dataset for intent classification and out-of-scope prediction. *EMNLP*.

Liu, A. et al. (2022). WANLI: worker and AI collaboration for natural language inference dataset creation. *Findings of EMNLP*.

Moharana, L. (2026a). calibrated-typed-decisions [dataset], versions v1.0, v2.0 and v2.1. huggingface.co/datasets/m-lagnajit/calibrated-typed-decisions.

Moharana, L. (2026b). Code for this report [software]. github.com/Lagnajit09/100x_AI_ML/tree/main/calibrated-typed-decision.

Moharana, L. (2026d). ctd-student-f and ctd-student-e [models]. huggingface.co/m-lagnajit/ctd-student-f, huggingface.co/m-lagnajit/ctd-student-e.

Moharana, L. (2026c). How sure is "sure"? Calibration, uncertainty, and knowledge in LLMs [blog post]. lagnajitmoharana.web.app/how-sure-is-sure-understanding-llm-calibration.

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology*, 12.

Ovadia, Y. et al. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *NeurIPS*.

Qwen Team (2024). Qwen2.5 technical report. *arXiv:2412.15115*.

Szegedy, C. et al. (2016). Rethinking the Inception architecture for computer vision. *CVPR*.

Warner, B. et al. (2024). Smarter, better, faster, longer: a modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv:2412.13663*.

Zhang, X., Zhao, J. and LeCun, Y. (2015). Character-level convolutional networks for text classification. *NeurIPS*.

Appendix A. Single students: mean (lowest–highest) over three seeds, 2 dials

Table 18. *Test and PubMedQA, ModernBERT-base.*

Run	test acc	test ECE	test LL	PubMedQA acc	PubMedQA ECE	PubMedQA LL
A	0.912 (0.909–0.917)	0.011 (0.006–0.019)	0.247 (0.230–0.267)	0.580 (0.538–0.614)	0.130 (0.097–0.171)	0.970 (0.945–0.997)
B	0.918 (0.913–0.923)	0.022 (0.016–0.026)	0.281 (0.268–0.289)	0.579 (0.568–0.585)	0.092 (0.020–0.149)	0.987 (0.957–1.005)
C	0.892 (0.886–0.898)	0.021 (0.020–0.023)	0.310 (0.307–0.313)	0.583 (0.576–0.592)	0.055 (0.027–0.106)	0.940 (0.926–0.952)
D	0.915 (0.909–0.918)	0.016 (0.015–0.019)	0.248 (0.241–0.256)	0.574 (0.548–0.595)	0.044 (0.028–0.064)	0.947 (0.878–1.017)
E	0.912 (0.908–0.915)	0.015 (0.014–0.016)	0.261 (0.252–0.267)	0.596 (0.591–0.605)	0.086 (0.048–0.107)	0.935 (0.924–0.946)
F	0.919 (0.917–0.921)	0.015 (0.015–0.016)	0.245 (0.241–0.249)	0.595 (0.584–0.607)	0.046 (0.019–0.068)	0.914 (0.888–0.953)

Table 19. *SNIPS, ModernBERT-base (seeds 1 and 2 retrained).*

Run	SNIPS acc	SNIPS ECE	SNIPS LL
A	0.949 (0.943–0.959)	0.028 (0.027–0.028)	0.155 (0.135–0.177)
B	0.948 (0.939–0.958)	0.033 (0.031–0.034)	0.175 (0.155–0.196)
C	0.951 (0.946–0.956)	0.018 (0.010–0.027)	0.152 (0.128–0.169)
D	0.953 (0.946–0.959)	0.010 (0.008–0.013)	0.137 (0.120–0.153)
E	0.960 (0.959–0.961)	0.010 (0.004–0.015)	0.125 (0.122–0.127)
F	0.950 (0.947–0.954)	0.016 (0.011–0.022)	0.139 (0.137–0.143)

Appendix B. Ensembles of three seeds

Table 20. *Ensembles, ModernBERT-base, 2 dials per member.*

Run	test acc	test ECE	test LL	SNIPS acc	SNIPS ECE	SNIPS LL	PubMed QA acc	PubMed QA ECE	PubMed QA LL
A	0.922	0.018	0.221	0.961	0.044	0.143	0.595	0.153	0.964
B	0.925	0.009	0.249	0.954	0.028	0.153	0.582	0.052	0.948
C	0.895	0.013	0.292	0.958	0.026	0.143	0.595	0.065	0.932
D	0.920	0.008	0.231	0.956	0.018	0.128	0.589	0.054	0.928
E	0.918	0.013	0.241	0.965	0.020	0.114	0.587	0.081	0.926
F	0.928	0.006	0.225	0.958	0.020	0.126	0.595	0.051	0.899

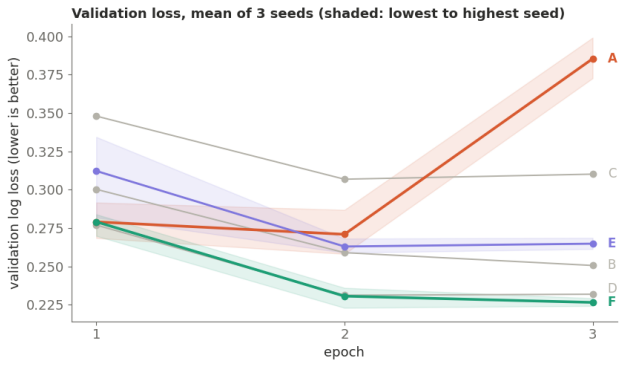


Figure 8. *Validation log loss per epoch, mean of three seeds (shaded: lowest to highest seed). Only A turns sharply upward in the third epoch.*

Appendix C. Kept epochs and compute

Table 21. *The epoch kept for each student (lowest validation log loss, out of 3).*

Students	seed 0	seed 1	seed 2
A (base)	1	2	2
B (base)	3	3	3
C, D, E (base)	2	2	3
F (base)	2	3	3
E and F (large, full fine-tuning)	3	2	3
F (large, LoRA)	3	2	2

Table 22. *Compute.*

Job	Hardware and cost
Teacher scoring	Qwen2.5-7B-Instruct in 4-bit on a free Colab T4; each question read once
Base student run	about 20 minutes on a T4
Large student, full fine-tuning	two pieces of 8 per step; about 0.6–0.7 s per step, about 40–45 minutes per run; 9.3 GB peak in a short trial
Large student, LoRA	about 0.58 s per step; one piece of 16 ran out of memory (10.6 GB already at step 20), two pieces of 8 were used
Analyses, card and inspections	CPU only

Appendix D. Configuration

All values below are read from the code: the builders and notebooks listed in Appendix F.

Table 23. *Dataset configuration.*

Dataset build	v1	v2 (and v2.1)
Teacher	Qwen/Qwen2.5-7B-Instruct, 4-bit NF4, 16-bit compute	same
System prompt	"You are a careful text classifier. Reply with only the letter of the single best option."	same
Questions per source (train / val / test)	1,000 / 300 / 300	3,000 / 300 / 300; SciCite 2,000 training
Where questions come from	train: source's training split; val and test: disjoint halves drawn from its evaluation split, generator seeded per source	same
Question types	30% yes/no; choice questions show 3–8 options	same
Wrong options	random	half from the 10 closest labels (bge-small-en-v1.5), half random
Option text	label	label: description (hand-written, or drafted by the teacher and reviewed)
Text length cap	1,200 characters	1,200; PubMedQA 2,000
Rows dropped	teacher's probability on option letters below 0.5	same
Teacher dials	per source and type, on validation; temperature grid 0.25–20 (200 log-spaced values); yes/no shift grid −8 to 8 in steps of 0.05; groups under 30 questions use a pooled per-type dial	same
Rater soft labels	—	GoEmotions vote shares + 0.1 pseudo-vote per option
Seed, license, tag	0 · CC BY 4.0 · v1.0	0 · CC BY 4.0 · v2.0, then v2.1 on main

Table 24. *Where each source was loaded from.*

Source	Loaded from	Configuration
CLINC150	clinc/clinc_oos	"plus"; evaluation split: test
Banking77	mteb/banking77 (license of PolyAI/banking77)	evaluation split: test
MASSIVE	mteb/amazon_massive_intent (license of alexa/massive)	"en"; evaluation split: test
GoEmotions	google-research-datasets/go_emotions	"simplified"; raw rater votes for soft labels
Amazon Polarity	mteb/amazon_polarity	evaluation split: test
SciCite	the authors' archive (train.jsonl, test.jsonl)	2,000 training questions
PubMedQA	qiaojin/PubMedQA (fallback: the authors' JSON)	"pqa_labeled", all 1,000 questions
SNIPS	sonos/nlu-benchmark, 2017-06-custom-intent-engines	validation files first, then full training files; 150 per intent

Table 25. *Student configuration.*

Students	v1	v2 (base and large)
Model	answerdotai/ModernBERT-base	ModernBERT-base; ModernBERT-large in Steps 8–9
Token caps (sequence / question / option)	512 / 64 / 24	512 / 96 / 32
Batch	16	16; large: two pieces of 8 with gradient accumulation
Epochs; kept epoch	3; lowest validation log loss	same
Optimizer	AdamW, weight decay 0.01, 10% warmup, clipping 1.0, fp16 autocast	same
Learning rates	body 3e-5, scorer 1e-3	same; LoRA adapters 2e-4
Target settings	B: smoothing 0.1	B: smoothing 0.1; D and F: one-hot share 0.5
LoRA	—	r 16, alpha 32, dropout 0.05, on Wqkv, Wo, Wi; peft inject_adapter_in_model
Randomness	seed fixes the scorer's start; batch order and option shuffles from random.Random(1000 · seed + epoch)	same; seeds 0, 1, 2
Dials	12 (per source and type)	2 (per question type), chosen in Step 1; same search grids as the teacher
Evaluation	10 ECE bins; 1,000 bootstrap resamples	same; SNIPS questions seeded separately (seed 0); 64-question weight check before reuse

Appendix E. Reproducing the results

Table 26. *Order of runs. Each notebook starts with SMOKE = True, a short end-to-end trial; set it to False for real runs.*

#	Run	Produces	GPU
1	build_ctd_v2.ipynb with SMOKE = False	dataset v2.0 on the Hub	yes (teacher)
2	update_ctd_v21.ipynb	dataset v2.1 (corrected labels)	no
3	student notebook, Cells 1–15, for SEED = 0, 1, 2	the six base runs, three seeds	yes
4	Cells 23, 24, 25	dial setups, seed tables, ensembles	no
5	Cells 26 and 27; then RERUN = True, SEED = 1 and 2, Cells 1–13, 26, 14, 15; then Cell 28	SNIPS for all three seeds	yes
6	MODEL_SIZE = "large", SEED = 0–2, Cells 1–13, 26, 14, 15; then Cell 30	large E and F	yes
7	also ADAPT = "lora"; then Cell 31	large F with LoRA	yes
8	Cell 29; update_card_results.ipynb	the figures; the card's results section	no
9	publish_students.ipynb	the two published students, their dials and model cards	yes

Appendix F. Artifacts

Table 27. *Code and results files.*

Artifact	Contents
build_calibrated_typed_decisions.py	builds and publishes dataset v1
build_ctd_v2.ipynb	builds and publishes dataset v2
update_ctd_v21.ipynb	adds the corrected-label column (v2.1)
train_calibrated_student_v2.ipynb	Cells 1–15: data, model and training; 16–22: single-run reports; 23: dial setups; 24: seeds; 25: ensembles; 26–28: SNIPS; 29: figures; 30: large students; 31: LoRA
update_card_results.ipynb	regenerates the card's results section from the results files, with safety checks
inspect_contractnli.ipynb, inspect_cosmos_qa.ipynb	the pre-registered dataset inspections of Section 6
publish_students.ipynb	packages, checks, measures and uploads the two published students
update_card_links.ipynb	the card's links and citation sections
m-lagnajit/ctd-student-f, m-lagnajit/ctd-student-e	the published students: weights for three seeds, dials, the ctd_student.py package, and model cards
step1_* to step9_* results files	every table in this report, saved as CSV next to the predictions