Any Language Model Is a Classifier: Calibrated One-Pass Decisions over Arbitrary Label Sets from Pretrained LLMs

Max Zhang
max.zhang@mail.mcgill.ca

September 21, 2026

Abstract

Many language-model calls select from a known set of options but use generation followed by parsing. We instead compute option likelihoods and normalize over the set. Packing the options after a shared prompt with a block attention mask computes all likelihoods in one forward pass (a pack-readout). Without training, this is the log-likelihood protocol of standard evaluation harnesses, with its known limitations: it favours short options, splits probability across paraphrases and is poorly calibrated. We train the option distribution by cross-entropy over the option set on a mixture of 18 tasks and evaluate on 12 held-out tasks. The mean primary score on the held-out tasks (macro-F1, or accuracy where options differ per example) rises from 0.37 to 0.57 at 0.6B parameters, from 0.55 to 0.64 at 1.7B and from 0.63 to 0.72 at 8B. A single set of weights handles every label set tested. Against the 35 zero-shot classifiers of the BTZSC leaderboard, it lies on or above the size–accuracy frontier on the three shared tasks; under the benchmark's own protocol, the 4B model exceeds the best listed system at half its size. Prompting the same-size instruction-tuned model is less accurate at every size; eleven prompted hosted models, including GPT-6 Astra and Claude Fable 5.1, score 0.71 to 0.81 against 0.72 for our trained 8B. A learned scalar head transfers better than the language-model head; at 1.7B, mean top-1 calibration error decreases from 0.114 to 0.098 without temperature scaling. A 77-way decision from a 1.7B model takes 54 ms on an A100, 3× faster than constrained decoding and 19× faster than JSON generation with a rationale; the conversion requires 16,000 training examples and 20 minutes of LoRA training and applies unchanged to SmolLM2, OLMo-2, Qwen2.5 and gpt-oss-20b. Model implementation, training code and trained adapters are released.

1 Introduction

In many applications, a language model selects one option from a fixed set: a routing label, a severity tier, yes or no. Generating text and parsing it has drawbacks. Decoding is sequential and memory-bound, so an n-token label requires n sequential forward passes; the output must be parsed and may fall outside the schema; and a sampled label carries no usable probability.

An alternative is maximum-likelihood selection among admissible options [5, 7]. Known options can be scored by teacher forcing without decoding. Restricting outputs to these options provides valid decisions with associated probabilities. Commercial "decision models" such as Jev [67] expose this interface (a typed question, a set of options, a probability over them, one pass), but do not disclose how they are trained. We investigate a simple, fully specified adaptation of a pretrained language model toward this interface: a classifier that accepts a new label set at inference time and returns probabilities over it in one forward pass.

Zero-shot option scoring has at least three documented limitations. Summed log-probabilities penalize long options and predictions change with the scoring method [66]; probability leaks to paraphrases outside

the option set [24]; and the renormalized distribution is poorly calibrated [27, 81, 83]. Per-task decoder methods train a classification head [38, 63]; Universal Discriminator [76] transfers encoder-based candidate scoring to held-out tasks. We combine cross-task discriminative training with pack-readout.

Our contributions are:

1. **Pack-readout.** One forward pass over the prompt followed by every option, with a block attention mask and per-option position ids, gives the same log-likelihoods as scoring each option separately, without KV-cache replication (Section 3.2); the same pass implements three decision primitives: choice, ordinal score and yes/no (Section 3.3).

2. **Training the option distribution.** We minimize cross-entropy over the option set on a public task mixture with at most 2,000 LoRA steps (16,000 example presentations, approximately 2.5M tokens); training takes 20 minutes on one A100 at 1.7B. On 12 held-out tasks, the mean primary score improves over zero-shot scoring by 9 points at 1.7B (three seeds), 4B and 8B, by 20 at 0.6B and by 2 at 14B with a reduced learning rate and step count (Section 5.2); the scalar head transfers to label sets and label names it never saw (Section 5.3).

3. **Comparison with dedicated zero-shot classifiers.** Against the 35 systems of the BTZSC leaderboard, the trained models are above every listed system on MASSIVE from 0.6B on and on Banking77 from 4B on (Figure 3); under the benchmark's own protocol, trained without its datasets, the 4B model is above every leaderboard system at half the size of the best one (Figure 8).

4. **Calibration.** The objective is a proper scoring rule: Brier score, log-likelihood and confidence ranking improve on unseen tasks, and top-1 calibration improves for the scalar head without temperature scaling (Section 5.4).

5. **Cost comparison against prompting.** A 77-way decision from a 1.7B model takes 54 ms on an A100, 3× faster than constrained decoding and 19× faster than JSON generation with a rationale, with little variation across inputs (Section 5.6). Prompting the instruction-tuned model of the same size is less accurate at every size, and up to 7% of its replies name no option; eleven hosted frontier models score 0.71 to 0.81 against 0.72 for our trained 8B, at 4 to 177 decode steps per decision (Section 5.5).

6. **A general recipe.** The same procedure raises SmolLM2-1.7B and OLMo-2-1B by 25 and 21 accuracy points, gpt-oss-20b by 15 and Qwen2.5-32B by 3; more training label sets help the many-label tasks and calibration most, and removing two training sets that neighbour held-out tasks raises the neighbours' scores (Section 5.7).

2 Background and Related Work

Option scoring. Scoring candidate continuations by log-likelihood is the multiple-choice protocol of GPT-3 [7] and lm-evaluation-harness [5], which reports raw (`acc`) and per-character normalized (`acc_norm`) variants. Holtzman et al. [24] identified surface-form competition and proposed domain-conditional PMI; Min et al. [46] score the prompt given the option; Tsvilodub et al. [66] show that predictions change with the scoring method. Scoring the letter of an option [57] is cheaper but biased toward particular option letters [82]; choices extracted from generated answers are more robust than first-token probabilities when the two disagree [70]. We score the full option text.

Discriminative fine-tuning of decoders. LS-LLaMA [38] and Sood and Savelka [63] attach a classification head to a decoder and train per task; the latter finds that cross-entropy over candidate scores outperforms

label-generation SFT by 2–3 points at 0.6–1.7B. Neither trains across tasks, evaluates on unseen label sets, or measures calibration or latency. Kowsher et al. [31] argue that generation preserves more information than a pooled representation; our LM-head logit is a token-level likelihood, the scalar head a pooled representation. We compare both with generation.

Zero-shot classification over arbitrary label sets. T0 [59] and FLAN [10, 72] train on many prompted tasks and transfer to held-out ones; both train to generate the target text, and T0 and the first FLAN evaluate classification by rank classification, the likelihood of each option (the later Flan collection evaluates by direct and chain-of-thought prompting). We adopt the held-out protocol and train the scoring readout directly. Universal Discriminator [76] trains an encoder-based scalar scorer across tasks with binary cross-entropy on input–option pairs and transfers to held-out tasks. We use a decoder, train with cross-entropy over the option set, and score all options in a shared-prefix pass. GLiClass [64] prepends the labels to the input of an encoder. BTZSC [1] benchmarks NLI cross-encoders, embeddings, rerankers and first-token LLM scoring on 22 datasets (Section 5.8). Laya [13], released concurrently, is an open encoder with Jev's three primitives: ModernBERT-large scores each option at a mask token and is trained per domain with REINFORCE on proper scoring rules. Its model card reports 0.36 zero-shot on its own decision set and 0.43 on Banking77, which indicates limited transfer to unseen label sets.

Shared-prefix attention. Hydragen [28] computes exact attention for sequences that share a prefix. Wang and Hegde [69] pack chosen and rejected responses behind a shared prompt with a block-sparse mask for direct preference optimization. We apply this construction to an option set, using likelihood or scalar readouts. Block-Attention [41] and Parallel Context Windows [56] mask independent blocks that a final block reads, and require fine-tuning or lose some accuracy. Pack-readout uses the same prefix-plus-block pattern for option scoring and needs no adaptation: each option block attends to only the prompt and itself.

Calibration. Temperature scaling [21], contextual calibration [81] and batch calibration [83] correct a fixed model's outputs; ConfTuner [36] trains verbalized confidence with a Brier loss. We train the option distribution itself with a proper scoring rule [20] and test whether calibration transfers. G-Eval [39] takes the probability-weighted sum of rating tokens; our ordinal primitive is the same estimator. Table 21 (Appendix C) compares these methods.

Concurrent work. Cheng et al. [9] concurrently introduce a 2B typed decision model that reads each option at a single token and is trained with full-parameter cross-entropy and Brier losses. We score the full option text, compare two readouts, and measure transfer to held-out tasks across model families with LoRA; Section 5.3 trains their single-token readout under our protocol.

3 Method

Figure 1 summarizes the model: a pretrained decoder run once over the prompt and all options, with a block attention mask and per-option position ids, read out at the option tokens and normalized over the option set.

3.1 Option scoring

Given a prompt x (state, question and the listed options) and options $o_1, \ldots, o_n$, each rendered as a leading space, the option text and a newline terminator, the score of option i is its teacher-forced log-likelihood and

the decision distribution its softmax:

$$s_i = \sum_{t=1}^{L_i} \log P_\theta\big(o_{i,t} \mid x,\, o_{i,<t}\big)\,, \qquad p_i = \frac{\exp(s_i/\tau)}{\sum_j \exp(s_j/\tau)}. \tag{1}$$

The terminator is scored so that an option whose tokens are a prefix of another's ("New York", "New York City") can score lower. A token prefix shared by several options contributes equally to their log-probabilities, leaving their relative odds unchanged while counting against options that do not share it. We also consider per-token, per-byte and per-character normalization of s_i, and the PMI variant $s_i - \log P_\theta(o_i \mid x_0)$ for the content-free prompt x_0 that keeps only the cue line, the domain-conditional PMI of Holtzman et al. [24].

3.2 Pack-readout: one forward pass for all options

Eq. 1 requires no decoding. A direct implementation prefills x once, replicates its KV cache n times and scores the padded option batch; the replication costs $O(nP)$ memory traffic and dominates for many options (Table 27). Instead we run one sequence $x \,\|\, o_1 \,\|\, \cdots \,\|\, o_n$, where $\|$ denotes token concatenation, with (i) a block attention mask in which a token attends to the prompt and earlier tokens of its own option only, and (ii) position ids that restart at $P = |x|$ for every option (Figure 1). The hidden states of option tokens then equal those of a separate pass over $x \,\|\, o_i$, so the log-likelihoods are exact (verified to 2×10^{-3} nats in fp32). Each option token's log-probability is read at the position that predicts it: the prompt's last position for the first token of every option, then the option's own positions, including the one that predicts its terminator; the LM head is applied only at these positions. We call this pass a pack-readout: the options are packed, then read out. The cost is one prefill of $P + \sum_i L_i$ tokens: an option adds L_i tokens to a compute-bound prefill, whereas each decoded token costs a memory-bound step.

3.3 Decision primitives

Three primitives cover the target decisions, each defined by Eq. 1. **Choice** lists the options in the prompt and returns p. **Score** lists K ordered levels with their indices, scores the index strings "0",…,"$K-1$", and returns the expected level $\sum_k k\, p_k$, as in G-Eval [39]; single-token indices ($K \le 10$) remove length bias from the ordinal case. **True/false** scores "yes" and "no" and returns p_{yes}.

3.4 Training the option distribution

We fine-tune with LoRA [26] on a mixture of classification and multiple-choice tasks, minimizing the negative log-probability of the gold option y under Eq. 1 with $\tau = 1$:

$$\mathcal{L}_{\text{set}} = -\log \frac{\exp(s_y)}{\sum_j \exp(s_j)}. \tag{2}$$

This trains the inference distribution with a strictly proper scoring rule, whose unrestricted population-risk minimizer is the true conditional distribution [20], and allows the model to correct length bias. We compare two readouts on the same packed computation: the *LM head*, s_i as in Eq. 1, and the *scalar head* of Sood and Savelka [63], $s_i = w^\top h_i$, where h_i is the final hidden state after the last norm at option i's newline terminator. This is the vector the LM head reads, computed with attention to the prompt and the whole option. Both readouts can score unseen option strings; p is the softmax over the n scores. We also compare the *gold-only* objective $-s_y$, which raises the gold option's likelihood without comparing it to the alternatives and is the label-generation objective of T0 and FLAN with the option text as target.

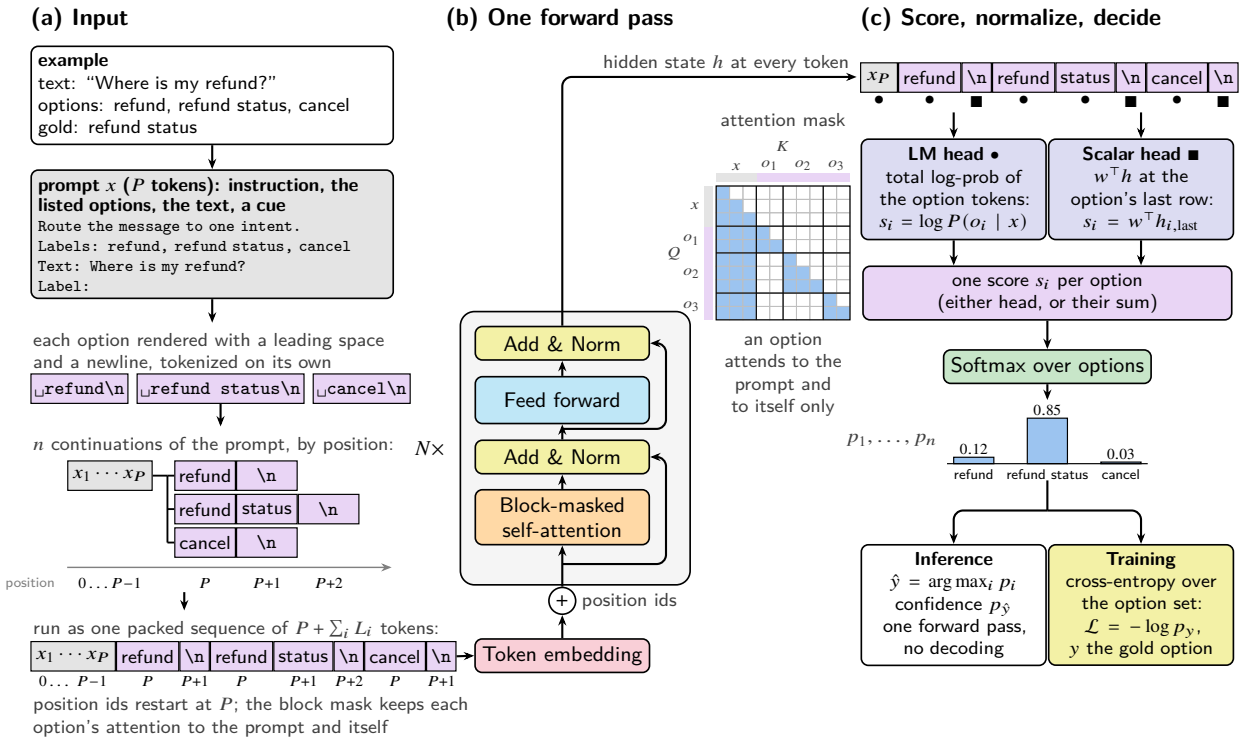


Figure 1: The model, on an intent example whose options include a prefix pair ("refund", "refund status"). (a) The example is formatted as a prompt that lists the options, and each option comprises a leading space, its text and a scored newline terminator, allowing a prefix option to score below its extension. Each option continues the prompt from position P, as if it were the only continuation; the n continuations are packed into one sequence for the forward pass. (b) A pretrained decoder runs once over that sequence. A block attention mask lets option tokens attend to the prompt and to their own option only, so each option's hidden states equal those of a separate pass over prompt and option. (c) Each option receives one score from its own positions: the LM head sums the log-probabilities of its tokens (read at the positions that predict them, •), the scalar head applies a learned vector to its last hidden state (■). A softmax over the n scores gives the decision distribution; training minimizes its cross-entropy against the gold option.

Is reinforcement learning needed for calibration? Jev is reported to be trained by "reinforcement learning for calibrated decisions" (RLCD), which its developers describe as optimizing for calibrated probabilities on decision tasks [68]; the reward and the data are not public. A reward that targets calibration is a proper scoring rule of p against a verified label. When the model emits the whole distribution p in one pass, such a reward, for instance the Brier score $\sum_i (p_i - \mathbb{1}[i = y])^2$, is deterministic and differentiable: there is no action to sample, gradient ascent on the expected reward is gradient descent on the loss, and a sampling estimator can only add variance. We therefore compare three objectives: the log score (Eq. 2), the Brier score as a loss, and REINFORCE [75]: sample a decision from p, then update the log-probability of that option in proportion to the reward minus a running mean of past rewards. With reward 1 when the sample is y, the expected reward is p_y rather than $\log p_y$, an improper scoring rule maximized by putting all mass on the likeliest option; it optimizes accuracy alone.

4 Experimental Setup

Models and compute. We train Qwen3-Base at 0.6B, 1.7B, 4B, 8B and 14B parameters [55] in bf16. Most runs use one NVIDIA A100-40GB; the main 8B and 14B runs use a DGX Spark (GB10, 128 GB), and the 14B sweep, the BTZSC ladder and the label-set sweep use a Lambda 8×A100-40GB with one run per GPU,

Table 1: Training mixture and held-out tasks. $|O|$: number of options; "per ex.": options differ per example. Sources: Appendix I.

role	task	$\|O\|$	kind
train	AG News, DBpedia-14, Yahoo Answers	4 / 14 / 10	topic
train	Yelp-5, IMDB, TweetEval sentiment	5 / 2 / 3	sentiment, ordinal
train	Emotion (dair-ai), TweetEval offensive, irony	6 / 2 / 2	emotion, binary
train	CLINC-150	151	intent
train	SciQ, OpenBookQA, CommonsenseQA, QASC, ARC-Easy, HellaSwag, WinoGrande	per ex.	multiple choice
train	BoolQ	2	yes/no with passage
held out	Banking77, MASSIVE	77 / 59	intent
held out	SST-5, Rotten Tomatoes	5 / 2	sentiment, ordinal
held out	TweetEval emotion, hate; GoEmotions	4 / 2 / 28	emotion, binary
held out	SMS Spam, Amazon Counterfactual	2 / 2	binary
held out	ARC-Challenge, MMLU	per ex.	multiple choice
held out	StrategyQA	2	yes/no

except gpt-oss-20b and Qwen2.5-32B, each sharded over two GPUs. Training uses LoRA of rank 16 ($\alpha = 32$) on all attention and MLP projections, AdamW [40] with learning rate 2×10^{-4}, 100 warm-up steps and cosine decay, and batches of 8 examples for 2,000 steps (16,000 examples of the 36,000-example mixture, at most 2,000 per training task); the scalar head is one fp32 vector trained with the adapter, and the main runs use seeds 0–2 (Appendix G describes the 4B seeds). A 1.7B run takes about 20 minutes of training and 3 minutes of evaluation on the subsets (12 on the full test sets). Training and evaluation jobs total 77 GPU hours in our records (13 on the Colab A100, 12 on the DGX Spark and 52 on the Lambda A100s), mostly for the 14B and larger models, the two sweeps of Section 5.7, and the prompting and latency benchmarks; the records omit the full-test evaluations, whose logs were not kept.

Tasks. Table 1 lists the mixture and the held-out set. Held-out tasks share no dataset and no label set with the mixture, but two have close neighbours in it: three of the four TweetEval emotion labels (anger, joy, sadness) also appear in the six-label Emotion training set, which lacks "optimism", and TweetEval hate is closely related to TweetEval offensive, a broader category. Section 5 reports the effects; Section 5.7 evaluates training without these datasets. Options are always listed in the prompt. Classification prompts read "*instruction* / Labels: …/ Text: …/ Label:"; question prompts read "Question: …/ Options: …/ Answer:" (Appendix A).

Baselines. At the same base model: zero-shot option scoring with raw, per-byte and PMI scores and with batch calibration; five-shot option scoring, five training examples of the task in the prompt and no training; first-token symbol scoring, the LLM protocol of BTZSC; gold-only SFT and the scalar head on the same mixture; the scalar head trained on each held-out task's own training split, the supervised ceiling of the readout (Table 3). External baselines are the family leaders and small LLMs reported by BTZSC, and the best published zero-shot and supervised results per held-out task (Appendix E).

Metrics. Macro-F1 for tasks with a shared label set and accuracy for the two whose options differ per example (the *primary* metric); negative log-likelihood, Brier score [6] and expected calibration error [52] (ECE, 15 equal-width bins) of the top-1 confidence; AUROC of the confidence as an error detector; accuracy at 50% coverage. Macro-F1 averages per-class F1 over the classes present in the gold labels of the evaluated examples; a reply naming no option counts as an error for its gold class only. Latency is the wall time of one decision, measured once per input over 20 inputs after warm-up, on A100-40GB GPUs with the same eager PyTorch setup for every method and size (Appendix F names the hosts). Table 3 evaluates on the full

Table 2: Zero-shot held-out accuracy, Qwen3-1.7B-Base: raw score, per-byte normalization, PMI, batch calibration (+BC), first-token symbol scoring, and ECE of the raw score. Per-token and per-character normalization behave like per-byte.

task	raw	token	byte	char	PMI	+BC	symbol	ECE (raw)
banking77	0.432	0.228	0.198	0.174	0.014	0.478	–	0.083
massive	0.656	0.592	0.592	0.578	0.056	0.666	–	0.047
sst5	0.428	0.298	0.304	0.296	0.244	0.384	0.292	0.164
tweet emotion	0.758	0.758	0.678	0.620	0.690	0.762	0.734	0.078
tweet hate	0.612	0.588	0.588	0.588	0.588	0.622	0.676	0.062
rotten tomatoes	0.872	0.872	0.872	0.872	0.730	0.898	0.854	0.062
sms spam	0.796	0.796	0.634	0.506	0.878	0.590	0.236	0.162
counterfactual	0.824	0.827	0.827	0.827	0.827	0.588	0.827	0.107
go emotions	0.232	0.230	0.162	0.152	0.186	0.258	–	0.316
arc challenge	0.734	0.722	0.730	0.724	0.512	–	0.716	0.058
mmlu	0.584	0.566	0.556	0.558	0.430	–	0.548	0.128
strategyqa	0.582	0.582	0.608	0.608	0.510	0.634	0.512	0.102
mean	0.626	0.588	0.562	0.542	0.472	0.588	0.599	0.114

test sets (40,117 examples over the twelve tasks); the ablations, sweeps and other sizes use 500 seeded test examples per task, and Tables 11 and 12 check these subsets against the full test sets at every size (means within a point, except the 0.6B zero-shot mean, 1.2 lower on the subset). Bootstrap 95% intervals on the subsets have a half-width of about 0.04 per task (median; 0.07 at most, on the 335-example Counterfactual set) and 0.012 on the held-out mean. Per-task differences of a few points between single runs are within noise, so we compare means. Held-out means weight tasks equally; standard deviations describe variation across training seeds.

5 Results

5.1 Zero-shot option scoring

Our scorer reproduces lm-evaluation-harness on ARC-Easy (first 400 test questions, Qwen3-0.6B-Base: `acc` 0.6275 for both; `acc_norm` 0.560 for both). Table 2 gives zero-shot held-out accuracy for Qwen3-1.7B-Base. Text scoring is a strong baseline (mean 0.626; 0.73 on ARC-Challenge, 0.87 on Rotten Tomatoes) and, on the nine tasks with at most 26 options, 8.9 points above first-token symbol scoring (0.688 vs. 0.599), which fails on SMS Spam (0.24). Four observations motivate training. No normalization wins everywhere: the raw sum is best on average, and every length normalization reduces accuracy on short labels (Banking77: 0.43 raw, 0.17 per character). PMI with a generic null prompt collapses on large label sets (0.01 on Banking77): the option with the rarest surface form is predicted for every example. Small models follow the label prior: the 0.6B model predicts "spam" for 499 of 500 messages and "hate" for 494 of 500 tweets; batch calibration [83] improves the tasks dominated by the label prior (+4.6 on Banking77, +5.2 on StrategyQA at 1.7B) and degrades the imbalanced ones (−20.6 on SMS Spam, −23.6 on Counterfactual). Calibration is poor and inconsistent: ECE spans 0.05–0.32 across tasks at 1.7B (0.07–0.73 at 0.6B), and the best-fitting temperature spans 0.4–2.4 at 1.7B (0.5–20 at 0.6B, where 20 is the ceiling of the search), so no single post-hoc temperature suffices.

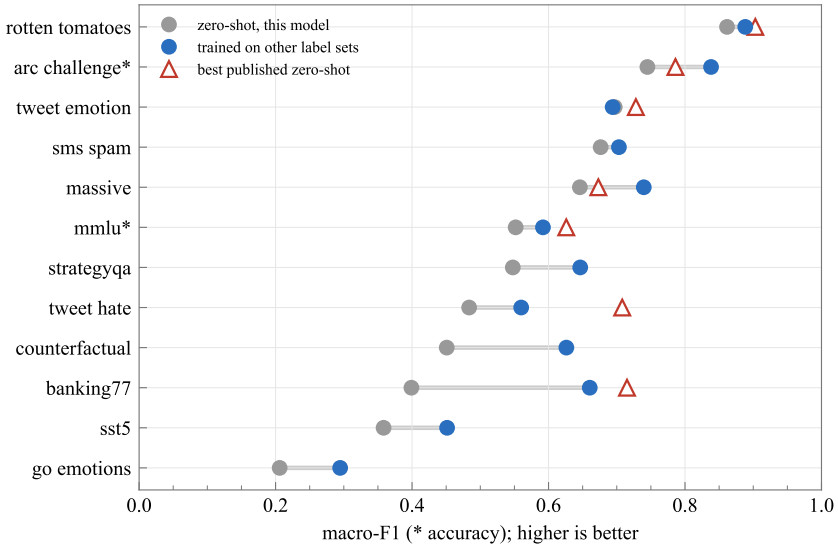


Figure 2: Before and after training, per held-out task, Qwen3-1.7B-Base, on the full test sets. Grey: zero-shot option scoring; blue: after training on the mixture (scalar head), which never saw these datasets. Open triangles: the best published zero-shot result on the same metric, from dedicated classifiers or earlier LLM reports of 1.7B to 20B parameters or of undisclosed size (Table 3 adds the supervised and prompted-frontier numbers). Tasks ordered by zero-shot score.

5.2 Transfer to unseen label sets

Training on the mixture raises the held-out mean on the full test sets (Table 3; Figure 2) from 0.552 to 0.621 with the LM head and to 0.641 ± 0.010 (three seeds; population s.d.) with the scalar head at 1.7B; five in-context examples of the task itself reach 0.571 (Table 13), so cross-task training yields about five times the gain of within-task prompting. The gain is largest where zero-shot scoring is weakest: 20 points at 0.6B (0.37 to 0.57), and at 1.7B Banking77 (77 unseen intents) from 0.40 to 0.66, Counterfactual from 0.45 to 0.63 and SST-5 from 0.36 to 0.45. The gain is 9 points at 4B and 8B and under one point at 14B (0.708 to 0.714). The 14B base model starts high: half of its 8-point zero-shot lead over the 8B base reflects a label prior aligned with three tasks (with batch calibration, which removes the prior, the zero-shot means are 0.58, 0.61 and 0.65 at 4B, 8B and 14B). The fixed 2,000-step run also produces regressions on short tweets that offset its gains on many-label tasks (Appendix B). Section 5.7 shows that a gentler run recovers the loss (0.731) and attributes the tweet regressions to two training sets that neighbour them. Accuracy and macro-F1 move differently on the imbalanced binary tasks: on Counterfactual the zero-shot model predicts the majority label for 334 of 335 examples (accuracy 0.82, macro-F1 0.45) and the trained model predicts both classes (0.84, 0.63). Training attenuates the prompt's label prior in favour of the text.

Against published numbers. The last columns of Table 3 report the best published zero-shot and supervised results on the same metric, with model sizes (Appendix E), and the best prompted frontier model from Section 5.5; Figure 2 marks the published zero-shot results. The frontier models exceed every published zero-shot number except on SST-5, defining the current zero-shot ceiling; our trained 8B trails the best model (Claude Fable 5.1, 0.806) by 8.5 points on the held-out mean, with 8B parameters against models of undisclosed or trillion-parameter size. The published zero-shot leaders are 8B embedding and reranker models, Flan-UL2 (20B) and ChatGPT or GPT-4, 5 to 12 times our model's parameters where the size is known. On MASSIVE, our 1.7B model, which never saw its 59 intents, reaches 0.74 against 0.67 for the 8B reranker; on Banking77 it reaches 0.66 against 0.72 for the 8B embedding model, and models trained on the

task's own labels remain well ahead there (0.94 accuracy with BERT-large). Both are on the full test sets; the published results use each paper's prompts and label wording, so one-point differences do not establish rankings.

Against the BTZSC leaderboard. Figure 3 places the model among the 35 zero-shot classifiers of the BTZSC leaderboard on the three held-out tasks the benchmark also evaluates, with, for each family, the best score at or below each size. One set of weights serves every label set, and none of our models was trained on the evaluation task. On MASSIVE the trained 1.7B model (0.74) is above every leaderboard system, whose best is an 8B reranker at 0.67, and the 0.6B model ties that reranker at 13× fewer parameters. On Rotten Tomatoes the 1.7B model is within two points of the best system and the 8B model above it. On Banking77, where embedding models that retrieve intents by similarity are the specialists, the 1.7B model lies just below their best score (0.66 against 0.72) and the 4B model above every system (0.73). Every LLM on the leaderboard, scored by the first-token protocol, is more than 20 points below ours on the intent tasks at up to 12B parameters. The four other base models (lettered discs) fall on or above the Qwen3 curve; gpt-oss-20b exceeds every system on Banking77 and MASSIVE, and Qwen2.5 32B every system on Rotten Tomatoes. The comparison is not exact: leaderboard numbers use the benchmark's verbalized label descriptions, ours the label names; Table 30 (Appendix D) gives our numbers under the benchmark's own protocol.

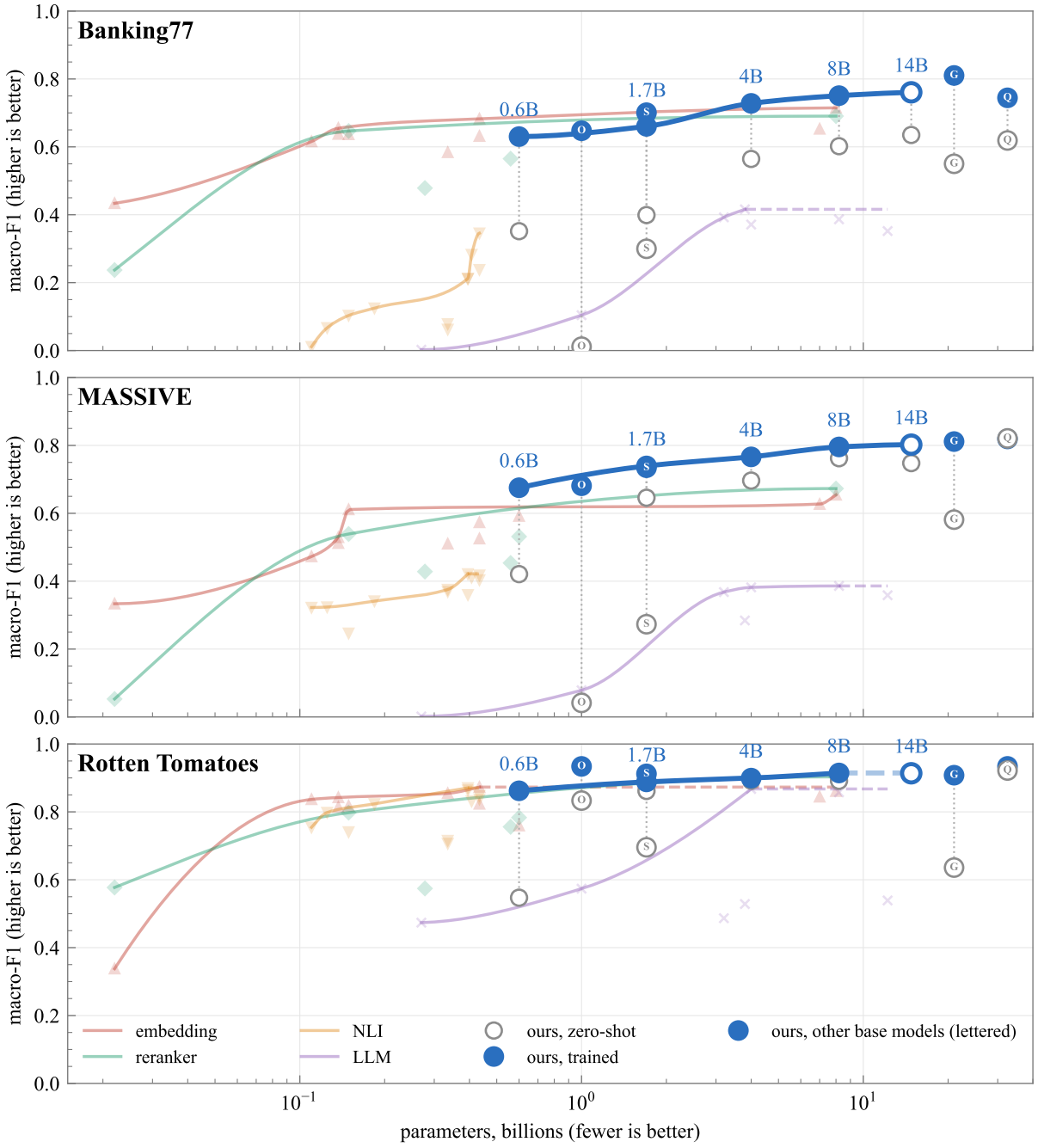
Size and compute per decision. Table 22 (Appendix C) estimates the cost of one decision for each of these systems as 2× parameters × tokens processed. The packed pass processes the prompt and every option once, 2.4 TFLOPs on Banking77 and 0.2 on SST-5 at 1.7B; a cross-encoder reranker runs one pass per label and costs 8× more on Banking77, and a 20B generative model 6× more before it decodes. A bi-encoder that caches the label embeddings costs less on Banking77, but it returns one similarity per label, each scored without seeing the others. Among systems that condition on the whole option set, ours is the smallest and costs no more per decision than any other.

How far from a model trained on the task? The "trained on the task" column of Table 3 runs the same recipe on each task's own training split for 400 steps. On the many-label and emotion tasks the gap is large: 0.87 against 0.66 on Banking77, 0.84 against 0.74 on MASSIVE, 0.85 against 0.69 on TweetEval emotion, 0.89 against 0.63 on Counterfactual. On ARC-Challenge and Rotten Tomatoes the general model is within two points or above (0.83 against 0.84, 0.91 against 0.89), on SST-5 and StrategyQA it trails by 11 and 6 points, and on TweetEval hate the task-trained model is worse (0.42 against 0.56): 400 steps on the imbalanced binary training split overfit. On the question and review tasks the general model is close to the task-trained one; elsewhere task-specific training adds 6 to 26 points, most where the label set is large and specific.

Gold-only SFT reaches 0.605 with the same readout and data (Appendix B): comparing the gold option against the alternatives adds about two points of macro-F1, with larger distributional improvements (Section 5.4). First-token symbol scoring stays below text scoring on every task where it applies, except TweetEval hate, where the letter prompt avoids the "hate" prior.

5.3 Which readout transfers?

In most cases, the scalar head transfers better than the LM head. At 1.7B it leads by 2.0 points of mean primary score over three seeds (0.641 vs. 0.621, Table 3), by 13 points on Counterfactual, by 2–3 on MASSIVE, SST-5, SMS Spam, GoEmotions and StrategyQA and by 1 on Banking77, while the LM head leads by about a point on MMLU and TweetEval emotion and the rest are within a point. It is also more robust to renaming. Replacing every label of the seven held-out tasks with small label sets by a paraphrase ("ham" to "legitimate



Lettered discs, our other base models (one size each): O OLMo-2 1B, S SmolLM2 1.7B, G gpt-oss-20b, Q Qwen2.5 32B

Figure 3: Macro-F1 against parameters on the three held-out tasks that BTZSC also evaluates. Faded lines: for each family of the BTZSC leaderboard [1], a smooth curve through its Pareto frontier (the best score at or below each size), ending at the last improvement, with a dashed continuation to the family's largest system; faint markers: the individual systems as published (`btzsc/btzsc-results`, fetched 2026-09-20; full test sets, verbalized label descriptions). Blue: our trained model at 0.6B to 14B (scalar head, label names, full test sets, Table 12; three seeds at 1.7B; outline only: the gentler 14B run of Section 5.7), which never saw these datasets; open circles: the same base models zero-shot; lettered discs: the four other base models of Section 5.7, one size each (O: OLMo-2 1B, S: SmolLM2 1.7B, G: gpt-oss-20b at 21B total parameters, Q: Qwen2.5 32B; 500-example subsets), open when zero-shot. A dotted segment joins each base model's zero-shot point to its trained point.

Table 3: Held-out performance, Qwen3-1.7B-Base: macro-F1, or accuracy for the two tasks marked † whose options differ per example. The LM-head and scalar-head columns never saw these datasets (both readouts: mean of three seeds); "trained on the task" is the same recipe run on the task's own training split (400 steps, scalar head), the supervised ceiling of this model. The last three columns are context: the best of the eleven prompted frontier models of Table 7 on the 500-example subsets (free replies), and the best published numbers on the same metric, on full test sets, from dedicated classifiers or earlier LLM reports not trained on the task (zero-shot) or trained on it (supervised), with the model's parameter count (n.d.: not disclosed); sources and non-comparable cases in Appendix E. Our columns are on the full test sets, three seeds for both readouts (Table 11 gives the 500-example subsets used elsewhere). Ablation columns are in Appendix B.

task	zero-shot	LM head	scalar head	trained on the task	best prompted frontier (model)	best published zero-shot (size)	best supervised (size)
banking77	0.399	0.650	0.660	0.874	0.868 (Claude Fable 5.1)	0.715 (8B)	–
massive	0.646	0.716	0.740	0.836	0.890 (GPT-6 Astra)	0.673 (8B)	–
sst5	0.358	0.429	0.451	0.564	0.544 (Claude Fable 5.1)	–	–
tweet emotion	0.697	0.701	0.694	0.849	0.802 (gpt-oss-120b)	0.728 (n.d.)	0.810 (0.12B)
tweet hate	0.484	0.562	0.560	0.419	0.767 (Kimi K3)	0.708 (20B)	0.581 (0.12B)
rotten tomatoes	0.862	0.883	0.888	0.911	0.952 (Claude Fable 5.1)	0.903 (8B)	–
sms spam	0.676	0.673	0.703	–	0.986 (Claude Fable 5.1)	–	–
counterfactual	0.451	0.498	0.626	0.890	0.847 (gpt-oss-120b)	–	–
go emotions	0.206	0.270	0.295	0.468	0.362 (gpt-oss-120b)	–	–
arc challenge†	0.745	0.837	0.838	0.828	0.976 (Kimi K3)	0.786 (3.2B)	–
mmlu†	0.552	0.605	0.592	–	0.938 (GPT-6 Astra)	0.626 (1.7B)	–
strategyqa	0.548	0.628	0.646	0.708	0.923 (Claude Fable 5.1)	–	–
mean	0.552	0.621	0.641				

Table 4: Readout comparison, Qwen3-1.7B-Base, seed 0: accuracy on all held-out tasks, on the seven whose labels can be renamed, and on the same seven with every label paraphrased. The single-token readout is measured on the 5,835 matched examples of Table 24, where the scalar head scores 0.696 and the LM head 0.650; it was not evaluated with renamed labels.

readout (seed 0)	held-out mean	7 renamable tasks	same, labels renamed
zero-shot	0.626	0.696	0.596
scalar head	0.698	0.730	0.661
LM head	0.661	0.679	0.586
both heads	0.644	0.648	0.600
LM head, byte-normalized logits	0.678	0.713	–
single-token readout	0.679	–	–

message", "optimism" to "hopeful"; Appendix A) costs the scalar head 6.9 points of accuracy, the LM head 9.3 and the zero-shot model 10.0 (Table 4). Both readouts depend on pretrained representations that need not be invariant to label paraphrases. This sensitivity is already present before fine-tuning (Section 5.7) and may reflect lexical familiarity or benchmark exposure during pretraining; this experiment cannot distinguish these explanations. At 0.6B the ordering is the same (0.568 vs. 0.527). Training both heads and summing their scores does not help: 0.617 held-out at 2,000 steps, below either head alone, and 0.600 on the renamed labels. These results favour the scalar head for training and the LM head for zero-shot scoring.

Single-token readout. The concurrent this-that-model [9] reads its decision at one designated position. To isolate the readout, we train it under otherwise identical conditions: the option letters' logits at the answer position (the first-token protocol of Section 5.1), with the same LoRA, objective and mixture (Qwen3-1.7B-Base, seed 0). On the same 500 test examples per task it reaches 0.626 against 0.652 for the scalar pack-readout and 0.617 for the LM head (Table 24, Appendix C). Paired bootstrap 95% intervals over examples are +1.5 to +3.8 points for the scalar head over the single-token readout and −2.2 to +0.4 for the LM head; the scalar head is also better calibrated (Brier 0.414 against 0.447, ECE 0.097 against 0.116). These results come from one seed, and the intervals do not include variation across seeds.

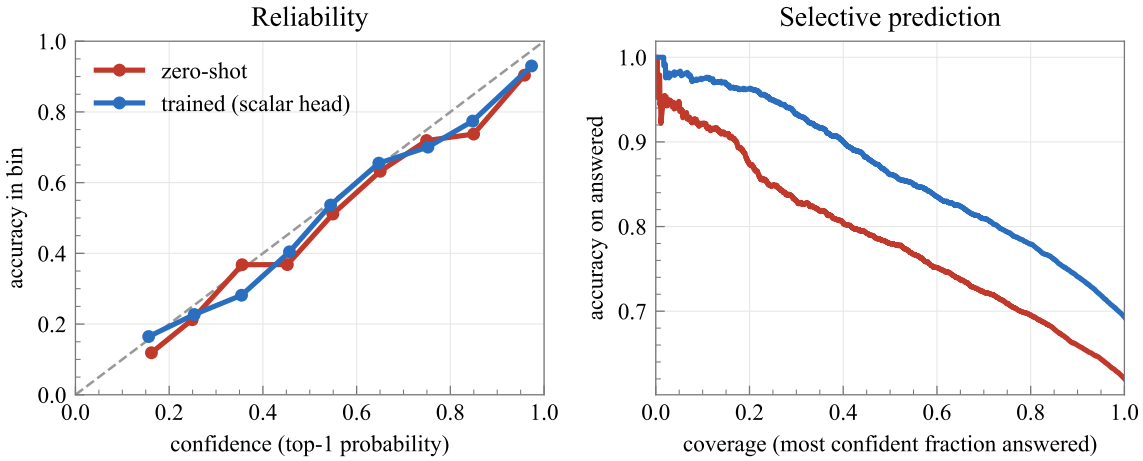


Figure 4: Calibration on held-out tasks, Qwen3-1.7B-Base, zero-shot vs. the trained scalar head, pooled over the twelve tasks (5,835 predictions; bins with at least 20). Left: reliability diagram; the dashed diagonal indicates equality between confidence and accuracy. Both models remain close to it without temperature scaling in this pooled view (pooled ECE 0.04 for both). Pooling can hide opposite errors on different tasks, so the text and tables report ECE per task, which is stricter (Table 5). Right: accuracy on the most confident fraction of examples; the curve rises as coverage shrinks because the confidence ranks correct decisions above wrong ones.

5.4 Calibration on unseen tasks

Table 5 and Figure 4 separate two notions of calibration. The proper scores that the objective optimizes improve on unseen tasks for both readouts: Brier from 0.507 to 0.452 (LM head) and 0.418 (scalar head), NLL from 1.149 to 0.994 and 0.881, together with confidence ranking (AUROC 0.69 to 0.73 and 0.75; accuracy on the most confident half 0.74 to 0.79 and 0.81). Top-1 ECE at $T = 1$ does not improve with the LM head (0.114 to 0.121) but does with the scalar head (0.098): the LM head's logits retain the language model's scale, improving confidence ranking without improving top-1 calibration; the scalar head learns its own scale. One cross-fitted temperature per task approximately halves ECE for both readouts, so a small labelled sample of a new task still improves its absolute calibration.

The objective matters more for the distribution than for accuracy. Gold-only SFT matches the log score on accuracy (0.664 vs. 0.667, LM head) but has the worst distribution of the models trained with a supervised loss (NLL 1.22 vs. 0.99; ECE 0.150 vs. 0.121): its loss never compares incorrect options with the gold one. Training on the Brier score, a calibration reward an RLCD-style method would optimize (Section 3.4), reduces accuracy by 2.9 points, improves ECE slightly (0.111) but produces a worse held-out Brier score (0.471 vs. 0.452) and AUROC (0.70 vs. 0.73); of the two proper scores, the log score is the better training objective. REINFORCE with an accuracy reward matches the log score on accuracy (0.657) but has a mean confidence of 0.92 at accuracy 0.66; its ECE (0.266) is twice the untrained model's and its NLL (2.45) twice that of any proper-score objective. This is the reward-induced over-confidence that Damani et al. [14] report for generators trained by reinforcement learning with verifiable rewards (RLVR) [32], reproduced in a one-pass decision model. When labels are available, the proper-score objective attains the same accuracy without this cost.

Ordinal decisions use the whole distribution. SST-5 has five ordered levels. Table 23 (Appendix C) compares the argmax with the expected level $\sum_k k\, p_k$, the Score estimator (Section 3.3). For all five models, the rounded expected level reduces mean absolute error and errors of two or more levels, and improves sentence ranking. The trained 1.7B scalar head improves most: mean absolute error falls from 0.64 to 0.55

Table 5: Calibration on held-out tasks, Qwen3-1.7B-Base, mean over tasks. "+TS": one temperature fitted on half of each task's test set, evaluated on the other half.

model	acc	ECE	Brier	NLL	AUROC	acc@50% cov.
zero-shot raw	0.626	0.114	0.507	1.149	0.690	0.743
zero-shot raw + TS	0.626	0.052	0.482	1.063	0.691	0.744
set-CE, LM head	0.667	0.121	0.452	0.994	0.729	0.789
set-CE, LM head + TS	0.667	0.053	0.426	0.878	0.719	0.783
gold-only SFT	0.664	0.150	0.475	1.220	0.736	0.792
set-CE, scalar head	0.692	0.098	0.418	0.881	0.749	0.814
Brier loss	0.638	0.111	0.471	1.008	0.697	0.743
REINFORCE (accuracy reward)	0.657	0.266	0.586	2.450	0.712	0.770

levels, errors of two or more levels from 8.7% to 5.5%, and rank correlation with the gold level rises from 0.76 to 0.83 (three seeds). The distribution retains information discarded by the argmax.

5.5 Against prompting a language model to name the label

Prompted classification asks an instruction-tuned model for a label and parses its reply. Decision models restrict outputs to the option set, eliminating parsing failures and invalid labels [68]. Table 6 compares these approaches at matched model sizes. Instruction-tuned Qwen3-1.7B, 4B and 8B receive the scorer's prompt without its cue line, followed by an instruction to return a listed answer, in chat format with thinking disabled. Free replies are sampled with the settings the model's authors recommend (temperature 0.7, top-p 0.8, top-k 20; seed 0) on the full test sets, matched to options, and unmatched replies count as wrong. We also evaluate greedy decoding constrained to the option strings.

First, prompting is less accurate than scoring the same model's base version zero-shot on most tasks, and far below the trained model: a mean primary metric of 0.52 for free and for constrained replies, against 0.55 zero-shot and 0.64 trained. Second, the free replies are invalid 1.7% of the time on average and 6 to 7% on Banking77 and TweetEval emotion, even though the instruction asks for one of the listed answers exactly as written. Table 25 (Appendix C) gives examples: on Banking77 the model produces plausible intents absent from the list ("reverted top up" for "top up reverted", "freeze my account" for "compromised card"), on emotion tasks it answers with a synonym of a label ("angry", "sad"), and on the question tasks it explains instead of choosing. Constrained decoding removes the invalid replies but leaves the mean unchanged, consistent with the model keeping its preference for its own phrasing in the first tokens. Third, every prompted decision requires sequential decoding: 99 ms median for free replies and 63 ms constrained on a Lambda A100-40GB, against 47–54 ms for the packed pass on the Colab A100-40GB (Section 5.6); the ratio is approximate across the two hosts. At 4B and 8B the same pattern holds with a smaller gap: the prompted models reach 0.63 and 0.66 with 1% invalid replies at 128–131 ms, against 0.60 and 0.63 zero-shot and 0.69 and 0.72 trained; our trained 1.7B (0.64) is ahead of the prompted 4B, and our trained 4B (0.69) of the prompted 8B.

Larger models. Table 7 repeats the free-reply protocol for eleven hosted models (Appendix F). Their held-out means range from 0.710 (DeepSeek V3.2) to 0.806 (Claude Fable 5.1), against 0.721 for our trained 8B (two seeds; the trained 14B scores 0.714, and 0.731 with 1,000 steps at half the learning rate, two seeds). DeepSeek V3.2, an open-weight model of 671B parameters, scores below it (our 8B is ahead on 6 of the 12 tasks); the current frontier models, GPT-6 Astra (0.793), Claude Fable 5.1 (0.806), Muse Spark 1.3 (0.790) and Kimi K3 (0.778, 2.8T parameters), are 6 to 9 points above it; GPT-5 and Claude Sonnet 4.6, of the previous generation, are 3 to 5 points above it. At most 0.9% of their replies name no option. They return no probability, their replies must still be parsed, and each decision requires sequential decoding: about 4

Table 6: Prompting the instruction-tuned model of the same size vs. option scoring, Qwen3 at 1.7B, 4B and 8B, twelve held-out tasks (macro-F1, or accuracy for †); ours on the 500-example subsets, the prompted models on the full test sets. Free: sampled reply matched to an option; constrained: greedy decoding restricted to the option strings. Invalid replies name no option and count as wrong. Latency per decision at batch size one, median over tasks: prompting timed on the task prompts on a Lambda A100-40GB; ours read from the sweep of Table 27 on the Colab A100-40GB at each task's number of options, one packed pass whether zero-shot or trained (Section 5.6).

	ours 1.7B		prompted Qwen3-1.7B		ours 4B		prompted Qwen3-4B	
task	zero-shot	trained	free	constrained	zero-shot	trained	free	constrained
banking77	0.401	0.680	0.414	0.417	0.578	0.756	0.606	0.606
massive	0.673	0.746	0.518	0.518	0.728	0.772	0.698	0.697
sst5	0.341	0.455	0.252	0.255	0.393	0.508	0.369	0.369
tweet emotion	0.698	0.658	0.505	0.537	0.710	0.672	0.674	0.675
tweet hate	0.501	0.582	0.352	0.352	0.624	0.566	0.610	0.609
rotten tomatoes	0.870	0.903	0.859	0.861	0.887	0.915	0.905	0.902
sms spam	0.645	0.678	0.731	0.719	0.510	0.766	0.537	0.537
counterfactual	0.452	0.641	0.616	0.599	0.528	0.764	0.711	0.719
go emotions	0.194	0.273	0.125	0.130	0.199	0.282	0.282	0.284
arc challenge†	0.734	0.825	0.713	0.719	0.836	0.885	0.867	0.868
mmlu†	0.584	0.631	0.542	0.535	0.658	0.702	0.659	0.669
strategyqa	0.546	0.618	0.591	0.591	0.523	0.710	0.655	0.649
mean	0.553	0.641	0.518	0.520	0.598	0.691	0.631	0.632
invalid replies			1.7%	0.0%			0.8%	0.0%
median ms per decision	48	48	99	63	60	60	131	81

	ours 8B		prompted Qwen3-8B	
task	zero-shot	trained	free	constrained
banking77	0.610	0.755	0.581	0.583
massive	0.802	0.809	0.741	0.727
sst5	0.257	0.526	0.423	0.425
tweet emotion	0.734	0.661	0.691	0.706
tweet hate	0.593	0.563	0.704	0.702
rotten tomatoes	0.891	0.927	0.912	0.915
sms spam	0.692	0.879	0.771	0.774
counterfactual	0.470	0.796	0.620	0.620
go emotions	0.225	0.320	0.231	0.238
arc challenge†	0.884	0.927	0.910	0.916
mmlu†	0.680	0.760	0.715	0.715
strategyqa	0.685	0.726	0.676	0.680
mean	0.627	0.721	0.665	0.667
invalid replies			0.7%	0.0%
median ms per decision			128	84

steps for Qwen3-235B and DeepSeek V3.2, 6 to 138 on the closed ones and 177 for Kimi K3, which reasons before it answers, against one prefill.

Figure 5 compares the three approaches by model size. The trained curve exceeds the prompted curve at every tested size, but the gap narrows from 12 points at 1.7B to 6 at 4B and 8B. Trained performance plateaus from 8B to 14B; hosted prompted performance continues to increase, reaching 0.78 for gpt-oss-120b. The four other trained base models (lettered discs) follow the same curve despite different initial scores: SmolLM2 1.7B reaches the Qwen3-1.7B score from 25 points lower, OLMo-2 1B follows the curve at 1B, and gpt-oss-20b and Qwen2.5 32B extend it. These results do not establish performance at frontier

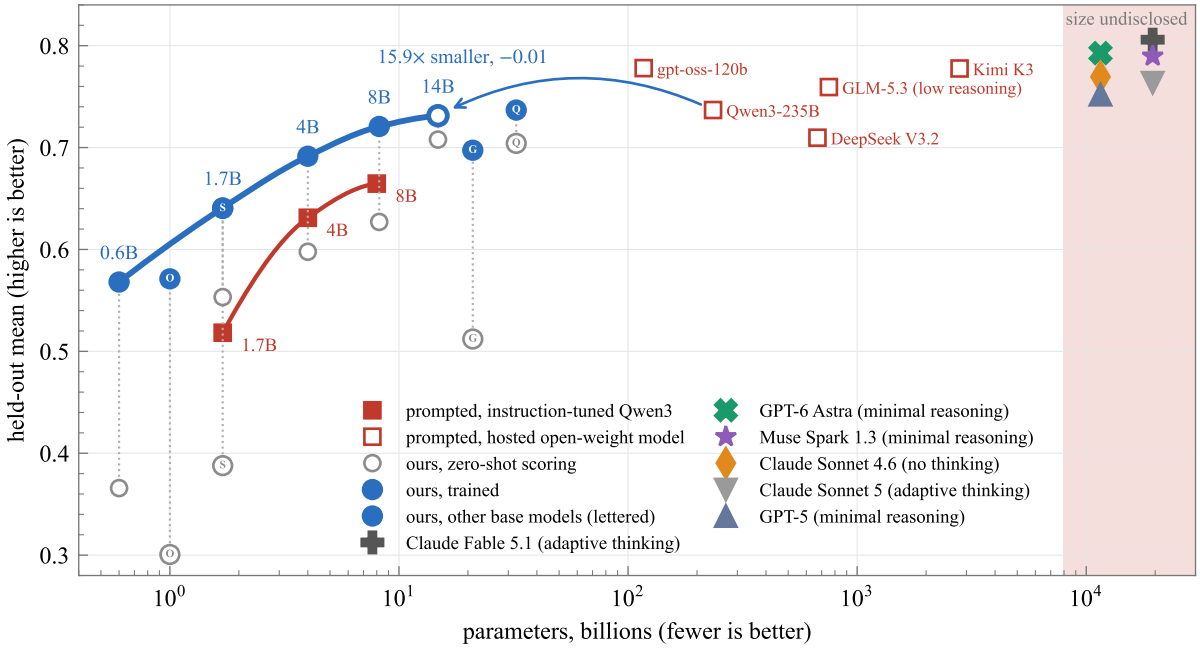


Figure 5: Held-out mean vs. parameters for three ways of using a language model as a classifier: prompting an instruction-tuned model to name the label (Qwen3 on our GPU; hosted open-weight models at their public size), zero-shot option scoring, and our trained models (Qwen3 at each size; outline only: the gentler 14B run of Section 5.7; lettered discs: the four other base models, one size each, gpt-oss-20b at its 21B total parameters). A dotted segment joins each base model's zero-shot point to its trained point; closed models of undisclosed size sit in the shaded column. Curves are smoothed Pareto frontiers (best at or below each size). The arrow compares our largest model with the prompted Qwen3-235B, the same family and pretraining, so the ratio is a within-family comparison (hosted sizes are total parameters; Qwen3-235B has 22B active).

scale; at the sizes tested, the trained model outperforms prompting at its own size and prompted models two to eighty times larger.

5.6 Efficiency

Figure 6 and Table 27 (Appendix C) compare the packed pass with KV-cache replication, one sequence per option, constrained greedy decoding of the label (a trie over option tokens) and JSON generation of a one-sentence rationale plus label, on the same base model and eager PyTorch execution. Absolute times are dominated by per-forward overhead in this setup (about 40 ms at 1.7B), so the ratios are the comparable quantity. Packed scoring uses the LM head; the JSON baseline adds an instruction to generate a rationale and label. Each method is timed once per input over 20 Banking77 queries and three random label sets per option count, so the bands of Figure 6 show variation across inputs and label draws. Packed latency is 47–48 ms from 2 to 16 options, 54 ms at 77, 91 ms at 255 (a 926-token prompt), with p90 within 2 ms of the median. KV replication costs about 2× throughout and 4.8× at 255 options, and at 4B does not fit in 40 GB at 255 options (Appendix H). Generation incurs a cost per decode step, with the total determined by the generated text: constrained decoding takes 164–217 ms median across the sweep, 2.4–4.3× the packed pass, with p90s up to 200 ms above its median. After prefill, each additional label token requires one decode step (3.0 to 3.7 steps on average over three random label sets per count), so decoding cost depends directly on the selected label's length; JSON with a rationale takes 1.0–1.2 s median, 13–22× the packed pass, with p90s 0.2–0.4 s

Table 7: Hosted instruction-tuned models prompted to name the label (free replies), against our trained 8B model: macro-F1, or accuracy for †, the share of invalid replies, and the mean number of decode tokens per reply, reasoning included. GPT-5, GPT-6 Astra and Muse Spark 1.3 run with minimal reasoning effort, GLM-5.3 with low effort, and the other models with their default settings, which for Claude Sonnet 5 and Claude Fable 5.1 means adaptive thinking on (Fable cannot turn it off) and for Claude Sonnet 4.6 means no thinking. Model ids, sizes, providers and settings in Appendix F.

task	ours, trained	Claude Fable 5.1 (adaptive thinking)	GPT-6 Astra (minimal reasoning)	Muse Spark 1.3 (minimal reasoning)	gpt-oss-120b	Kimi K3	Claude Sonnet 4.6 (no thinking)
banking77	0.755	0.868	0.857	0.833	0.843	0.847	0.800
massive	0.809	0.890	0.890	0.860	0.838	0.859	0.832
sst5	0.526	0.544	0.510	0.483	0.493	0.495	0.525
tweet emotion	0.661	0.771	0.775	0.769	0.802	0.758	0.761
tweet hate	0.563	0.701	0.605	0.753	0.734	0.767	0.752
rotten tomatoes	0.927	0.952	0.948	0.930	0.908	0.944	0.936
sms spam	0.879	0.986	0.976	0.958	0.897	0.911	0.930
counterfactual	0.796	0.818	0.821	0.821	0.847	0.736	0.721
go emotions	0.320	0.328	0.337	0.308	0.362	0.275	0.285
arc challenge†	0.927	0.968	0.972	0.970	0.950	0.976	0.964
mmlu†	0.760	0.922	0.938	0.924	0.862	0.912	0.898
strategyqa	0.726	0.923	0.883	0.868	0.800	0.851	0.829
mean	0.721	0.806	0.793	0.790	0.778	0.778	0.769
invalid replies		0.0%	0.0%	0.0%	0.1%	0.7%	0.1%
mean reply tokens		25	13	138	–	177	6

task	ours, trained	Claude Sonnet 5 (adaptive thinking)	GLM-5.3 (low reasoning)	GPT-5 (minimal reasoning)	Qwen3-235B	DeepSeek V3.2
banking77	0.755	0.811	0.784	0.787	0.733	0.726
massive	0.809	0.855	0.809	0.832	0.826	0.833
sst5	0.526	0.517	0.484	0.511	0.515	0.397
tweet emotion	0.661	0.741	0.736	0.748	0.768	0.761
tweet hate	0.563	0.676	0.741	0.669	0.697	0.544
rotten tomatoes	0.927	0.946	0.940	0.940	0.936	0.928
sms spam	0.879	0.973	0.920	0.868	0.862	0.907
counterfactual	0.796	0.564	0.718	0.801	0.737	0.689
go emotions	0.320	0.350	0.246	0.277	0.211	0.269
arc challenge†	0.927	0.970	0.974	0.958	0.950	0.942
mmlu†	0.760	0.908	0.894	0.876	0.826	0.848
strategyqa	0.726	0.854	0.868	0.757	0.782	0.674
mean	0.721	0.764	0.759	0.752	0.737	0.710
invalid replies		0.1%	0.1%	0.1%	0.9%	0.1%
mean reply tokens		13	13	21	4	4

higher. One sequence per option, the lm-evaluation-harness default, is competitive up to 8 options and then grows with options times prompt length. At on-demand A100 prices ($1.5–2 per hour), a 77-way decision costs about $0.03 per thousand.

Cost of the conversion. Converting a pretrained model requires one LoRA run: 2,000 steps of 8 examples, 16,000 presentations of public classification examples with 154 packed tokens each, 2.5M training tokens total, without synthetic data, labels from a larger model or reinforcement learning. At 6× parameters × tokens that is 25 PFLOPs at 1.7B and 120 at 8B, 20 minutes on one A100 at 1.7B (38 for the 1,000-step 4B run), 2.1 and 3.3 hours on a DGX Spark at 8B and 14B, with the time dominated by kernel-launch overhead. Table 26 (Appendix C) compares adaptation data and compute for dedicated zero-shot classifiers (Section 5.8) and instruction-tuned models. The best BTZSC system near our size, an NLI cross-encoder, was fine-tuned on about 885,000 NLI pairs for three epochs, roughly ten A100 hours for a comparable run; the best 8B system, Qwen3-Reranker-8B, on 19 million retrieval pairs, twelve million of them synthetic; the best embedding model on 800 million web-mined pairs and then 3 million annotated ones; Flan-T5-XXL's instruction tuning cost 3,000 times our run; and the post-training of the instruction-tuned and frontier models is not disclosed. Our conversion uses three orders of magnitude fewer examples than the retrieval-trained systems, every example is public, and it leaves the base model's weights untouched.

5.7 The recipe: base models and what drives transfer

Base models. Table 8 applies the identical procedure to Qwen3-Base at five sizes, SmolLM2-1.7B (Llama architecture, the size of Qwen3-1.7B), OLMo-2-1B (open training data), Qwen2.5-32B and gpt-oss-20b. SmolLM2-1.7B starts 19 points below Qwen3-1.7B zero-shot and finishes within a point (0.68 vs. 0.69 held-

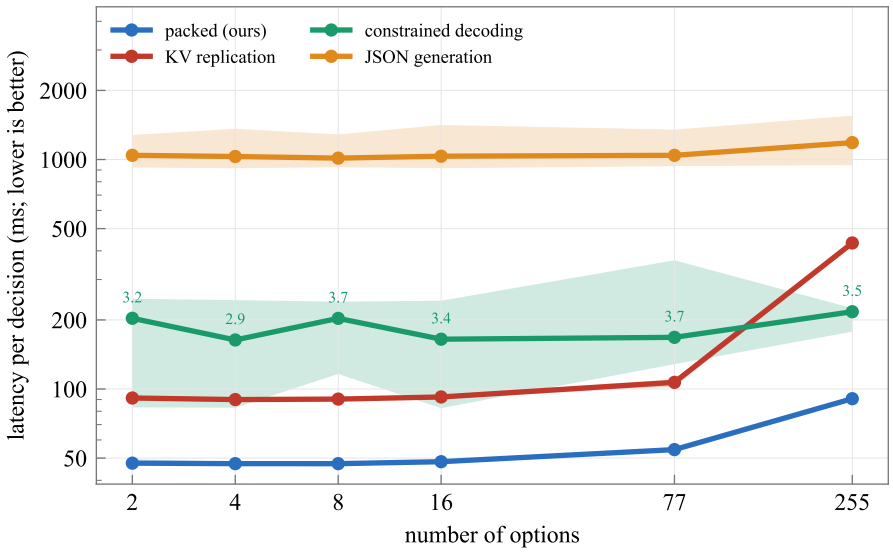


Figure 6: Latency per decision vs. number of options, Qwen3-1.7B-Base on an A100, both axes logarithmic: lines are medians over the same 20 inputs, bands the central 80% of inputs and label draws. Packed latency rises from 47–48 ms at 2–16 options to 91 ms at 255; its per-input spread is small. JSON generation with a rationale is an order of magnitude slower, while constrained decoding is 2.4–4.3× slower. Annotations report mean decode steps. Larger option sets may have longer labels, which lengthens decoding; each option count uses three random draws from pooled intent names, so this sweep does not separate the two effects.

Table 8: The recipe across base models: held-out mean accuracy and ECE ($T = 1$), zero-shot and after training with each readout.

base model	zero-shot	LM head	scalar head	ECE zero-shot	ECE LM head	ECE scalar
qwen3-0.6b	0.431	0.556	0.634	0.260	0.169	0.113
olmo2-1b	0.397	0.533	0.605	0.292	0.207	0.133
smollm2-1.7b	0.434	0.581	0.680	0.162	0.158	0.117
qwen3-1.7b	0.626	0.667	0.692	0.114	0.121	0.098
qwen3-4b	0.667	0.675	0.723	0.113	0.107	0.094
qwen3-8b	0.692	0.741	0.747	0.122	0.110	0.087
qwen3-14b	0.736	0.732	0.749	0.122	0.116	0.086
gptoss-20b	0.590	0.726	0.737	0.208	0.122	0.100
qwen2.5-32b	0.734	–	0.768	0.105	–	0.080

out accuracy); OLMo-2-1B, the weakest base, gains 21 points; Qwen3-4B and 8B gain 6 each and 14B one, all ending with ECE between 0.08 and 0.09: the accuracy gain shrinks as the base model grows, the calibration gain does not. gpt-oss-20b [51] differs from the others: a post-trained mixture of experts (21B parameters, 3.6B active) with sliding-window layers and attention sinks, run through a second, complete implementation of the method that builds its layer-wise masks. Zero-shot, it scores options poorly (0.51 macro-F1), as is typical of post-trained chat models; the recipe takes it to 0.70 with the scalar head, with ECE from 0.21 to 0.10, the largest gain of any model above 1.7B.

Training larger models. The recipe's learning rate and step count were chosen at 1.7B. At 14B the unchanged recipe fits the mixture more tightly (final training loss 0.35 vs. 0.44 at 8B) and gains only a point:

Table 9: Gentler runs at 14B: held-out mean, the three many-label tasks (Banking77, MASSIVE, GoEmotions), the four short-text tasks (SST-5, TweetEval emotion and hate, SMS Spam), and ECE. One seed each, two for the 1,000-step run at half the learning rate.

run	held-out mean	many-label tasks	short-text tasks	ECE
zero-shot	0.708	0.558	0.692	0.122
recipe: lr 2×10^{-4}, 2,000 steps	0.714	0.635	0.613	0.086
lr 2×10^{-4}, 1,000 steps	0.731	0.632	0.668	0.080
lr 10^{-4}, 2,000 steps	0.729	0.632	0.657	0.074
lr 10^{-4}, 1,000 steps	0.731	0.626	0.671	0.071
lr 5×10^{-5}, 2,000 steps	0.730	0.634	0.654	0.081
recipe, LM head	0.699	0.601	0.598	0.116

short-text regressions offset many-label improvements. Table 9 repeats the run with a lower learning rate, fewer steps, or both: every gentler run reaches 0.729 to 0.731, above the trained 8B, with the short-text tasks within two to four points of zero-shot, the many-label gains retained and ECE at 0.07 to 0.08. The amount of adaptation is decisive, whether controlled by step count or learning rate. The setting chosen at 1.7B, where increases in both continued to help, over-adapts a 14B model (the LM head at 14B falls below zero-shot on the mean, 0.699). Applied to Qwen2.5-32B, sharded over two GPUs, the same gentler run takes the held-out mean from 0.704 to 0.737, the highest of any trained model here, in 1.7 hours. The ladders of Figures 5 and 3 show this 1,000-step, half-learning-rate run at 14B as their outline-only point; the tables keep the unchanged recipe.

Number of label sets. At a fixed budget of 16,000 example presentations and 2,000 steps, we train the scalar head on 4 to 18 of the training tasks, three random subsets of tasks per count (Figure 7; Table 28 in Appendix C). Held-out accuracy increases with the number of training label sets at fixed data, with the largest increase on many-label tasks. Every count remains above zero-shot on the held-out mean (0.638 with 4 label sets against 0.626), which then rises to 0.69 at 18, with a spread across draws about as large as the trend. The three many-label tasks improve from 0.47 with 4 sets to 0.61 with 18, and ECE falls from 0.15 to 0.10. This is consistent with a model trained on few label sets learning those sets, and a model trained on many label sets learning to interpret the listed labels.

Amount of data. Table 29 (Appendix C) trains the scalar head for the same 2,000 steps on pools of 2,000 to 36,000 examples. Accuracy saturates early and calibration does not: the 2,000-example pool, seen eight times, already gives 0.595 against 0.553 zero-shot and the full pool 0.641, while ECE falls from 0.31 to 0.10. Repeated examples stop improving accuracy long before they stop increasing over-confidence, so the pool should be as large as the label sets allow.

Sensitivity to label names. Label names are part of the prompt and affect performance. On SMS Spam, "ham"/"spam" gives the 0.6B base model 0.12 accuracy (it calls nearly every message spam) and "legitimate message"/"unsolicited junk" 0.75; at 1.7B the order reverses (0.80 and 0.48), and the trained scalar head keeps the base model's preference (0.90 and 0.52).

Neighbouring label sets in the mixture. Two training sets closely resemble held-out ones (Section 4): Emotion shares three of TweetEval emotion's four labels, and TweetEval offensive is a broader version of TweetEval hate. Table 10 reports the same recipe without them. Removing these datasets improves their held-out counterparts: TweetEval emotion rises from 0.66 to 0.73 and TweetEval hate from 0.58 to 0.68

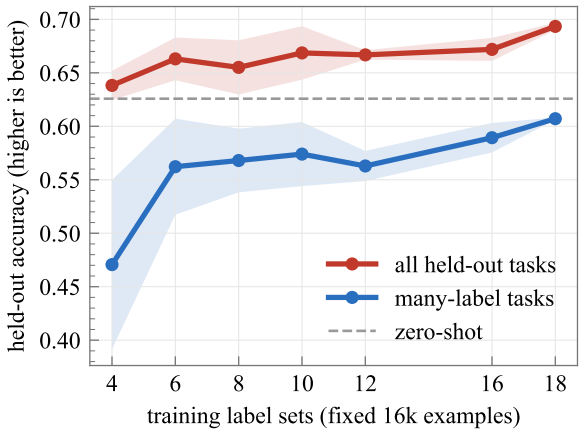


Figure 7: Held-out accuracy vs. number of training label sets at fixed data, Qwen3-1.7B-Base, scalar head; band: ±1 s.d. over three random task subsets per count. Accuracy increases with the number of training label sets at fixed data, most steeply on the many-label tasks; the mean is above zero-shot from four sets on.

Table 10: The mixture without the two training sets that neighbour held-out tasks (Emotion, TweetEval offensive), scalar head, 500-example subsets: the neighbours' held-out relatives improve and the held-out mean is unchanged. Each row is the mean over its seeds; the 4B pair are 1,000-step runs at the recipe's learning rate (the "short" setting of Table 9).

size	mixture	steps	seeds	TweetEval emotion	TweetEval hate	GoEmotions	held-out mean
1.7B	all 18 tasks	2,000	3	0.658	0.582	0.273	0.641
1.7B	without Emotion, TweetEval offensive	2,000	2	0.726	0.676	0.233	0.634
4B	all 18 tasks	1,000	1	0.732	0.566	0.291	0.693
4B	without Emotion, TweetEval offensive	1,000	1	0.749	0.618	0.264	0.698

(its ECE from 0.18 to 0.06), three to five times the seed spread. The held-out mean is unchanged (0.641 to 0.634): SMS Spam and Counterfactual, the two imbalanced binary tasks, lose about what the tweet tasks gain, within their seed spread; TweetEval offensive was the only imbalanced binary task in the mixture. At 4B, with 1,000-step runs, the same removal adds 2 points on emotion and 5 on hate, which returns to its zero-shot level, and again leaves the mean unchanged (one seed each). This suggests that the form of the label sets determines a training set's value to the mixture. A training set whose label set nearly duplicates a held-out one lowers that task's score, because the model learns a prior and class boundaries for labels that carry a different meaning at test time; a training set of the same form helps regardless of its domain.

What the training teaches. These results suggest that training teaches classification: interpreting the prompt's label set, comparing options and assigning probabilities. Six observations support this interpretation. Training uses 2.5M tokens of public labels against trillions of pretraining tokens, and accuracy saturates by a pool of 8,000 examples while the number of label sets continues to help (Tables 29 and 28). The gains occur where zero-shot scoring fails on form, the label prior and surface competition, and they are smallest for the base models that already handle the tasks (Section 5.2). With paraphrased labels, the scalar head still outperforms zero-shot scoring and the trained LM head (Section 5.3). The same procedure works on every tested base model, including a post-trained mixture of experts. Removing the two training sets that neighbour held-out tasks raises the neighbours' scores (Table 10), which is consistent with the mixture transferring form and a near-duplicate label set transferring a prior. The task-trained ceiling of Table 3 quantifies the additional value of task knowledge: 10 to 25 points on many-label tasks, where the general model relies on pretrained

label semantics. This hypothesis predicts that training on label sets from an unrelated domain should transfer nearly as well; testing it remains future work.

5.8 Comparison with dedicated zero-shot classifiers

BTZSC [1] evaluates zero-shot classifiers on 22 datasets from texts and a verbalized description of every label ("This example news text is about business news"). We list the descriptions in the prompt and score each as an option. Five BTZSC datasets (AG News, Yelp, IMDB, Emotion, Yahoo) are in our mixture, so we also train a model on the mixture without them. Figure 8 places our model among the 35 leaderboard systems by size; Table 30 in Appendix D gives macro-F1 overall and per task group against the best model of each BTZSC family and the small LLMs in its leaderboard, which are scored by the first-token protocol (0.43 overall for a 3B instruct model, 0.65 for Qwen3-4B).

On the two intent datasets shared with our held-out set, the overall-best model of each family scores 0.69 (reranker), 0.63 (embedding), 0.24 (NLI) and 0.35 (12B LLM) on Banking77 and 0.67, 0.57, 0.40 and 0.36 on MASSIVE; our 1.7B model reaches 0.66 and 0.74 with its own label names (Table 3). Under the benchmark's protocol, the untrained 1.7B base model scores 0.53 overall: above the 3B first-token LLM (0.43), below the 4B and 12B LLMs (0.65, 0.67) and below every specialist family, with the intent group (0.35) reduced most by the verbalized descriptions.

Trained on the mixture without the five BTZSC datasets, the same 1.7B model reaches 0.67 overall with the LM head and 0.70 with the scalar head: above every family leader except the 8B reranker (0.72), above the 12B first-token LLM (0.67) at a seventh of its size, and 27 points above the 3B LLM. By task group the scalar head is within 3 points of the reranker on sentiment (0.90 vs. 0.92) and emotion (0.48 vs. 0.49), 4 points below it on intent (0.66 vs. 0.70), and on topic (0.65) just below the reranker (0.66) and the 12B LLM (0.69). The same recipe at the other sizes gives 0.67 at 0.6B, 0.73 at 4B, 0.74 at 8B and 0.75 at 14B (the gentler run of Section 5.7; two seeds at 1.7B, 4B and 8B): from 4B on it is above the best system of the leaderboard, at half its size (0.727 vs. 0.722), and at the reranker's own size it is the best system on the leaderboard (0.740; intent 0.73 vs. 0.70, topic 0.69 vs. 0.66, emotion 0.45 vs. 0.49). Under the benchmark's own protocol, designed for classifiers that read a label description, the model trained on other label sets is above every family's frontier at each size we trained, on 500-example subsets of its datasets (Figure 8). Figure 3 compares individual systems on the three datasets shared with our held-out set. BTZSC reports inference speed only as a normalized figure without per-model numbers, on other hardware, so we make no speed claim against these families.

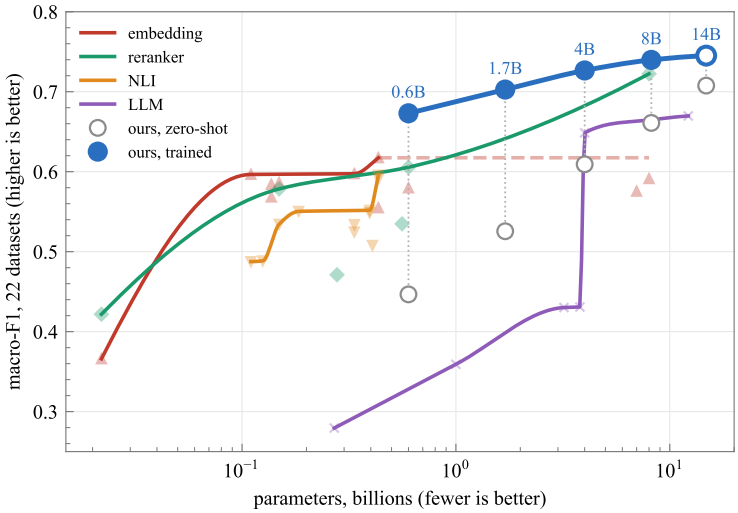


Figure 8: Overall macro-F1 on BTZSC (22 datasets, the benchmark's protocol) against parameters. Faint markers: the 35 leaderboard systems, on full test sets; lines: a smooth curve through each family's Pareto frontier, ending at the last improvement, with a faded dashed line to the family's largest system. Blue: our Qwen3 models trained on the mixture without the five BTZSC datasets (scalar head, 500-example subsets), at each size, with a curve through their frontier; the outline-only point is the 14B trained for 1,000 steps at half the learning rate (Section 5.7); open circles: the same base models zero-shot, a dotted segment joining each to its trained point.

6 Limitations

Two limitations follow from the design. Chain-of-thought prompting and reinforcement-trained reasoning raise accuracy on tasks that require intermediate steps [16, 73]; a one-pass decision has none, so the gain is unavailable to our model, and StrategyQA, the one held-out task that needs multi-hop inference, is among the tasks where our models trail the frontier most (0.65 at 1.7B and 0.76 at 14B against 0.92 for the best hosted model). For such questions, low-confidence cases could be routed to a reasoning model. The classifier interprets option text literally and therefore requires descriptive label names (Section 5.7).

The evaluation also has limitations. Held-out datasets are excluded from our fine-tuning mixture (two binary label sets recur, Section 4), but benchmark exposure during pretraining cannot be excluded. Our results therefore do not establish generalization to tasks absent from pretraining. Calibration is measured on the twelve held-out tasks with 500 examples each (335 for Amazon Counterfactual), and on TweetEval hate the trained model trades accuracy for macro-F1. All latencies come from one eager PyTorch implementation; speed ratios may differ under optimized serving. The recipe was tested with LoRA, on models of up to 32B parameters, in English, with hyperparameters tuned at 1.7B and only the gentler setting tested above 14B.

7 Conclusion

Selecting from a known option set requires no decoding. A pack-readout, one packed forward pass, computes exact option likelihoods at the cost of a prefill, and cross-entropy over the option set on public classification data turns a small pretrained model into a classifier that accepts a new label set at inference time: on 12 held-out tasks, the mean primary score rises from 0.55 to 0.64 at 1.7B and from 0.63 to 0.72 at 8B, while calibration improves (Brier 0.51 to 0.42 at 1.7B). One set of weights serves every tested label set: on unseen tasks it is on or above the size–accuracy frontier of dedicated zero-shot classifiers, leads their leaderboard from 4B on under their own protocol, and is more accurate than prompting the instruction-tuned model of

its size, at one prefill per decision. The method consists of a block attention mask, a softmax over the option set and one LoRA training run; it applied unchanged to all five base models tested. Model implementation, training code and trained adapters are released.

Acknowledgements and disclosures

Claude Code and OpenAI Codex assisted with training-code development, manuscript revisions, and figure-generation code. The research idea originated with the author, who takes full responsibility for the work. The study was entirely self-funded, and the author declares no competing interests.

Data availability

Benchmark sources are cited in Appendix I; the repository's `DATA.md` lists dataset identifiers and access links. The evaluation reports (per-example scores on the held-out tasks, aggregate scores on BTZSC, the hosted models' replies), the latency measurements and the BTZSC leaderboard snapshot are released as a results archive (https://github.com/maxz411/classifier-from-llm/releases/tag/v0.1.0).

Code availability

The model implementation and training and evaluation code are available under Apache-2.0 at https://github.com/maxz411/classifier-from-llm, as the Python package `packreadout`. The trained adapters are on Hugging Face under CC BY-NC 4.0 (https://huggingface.co/collections/maxz411/pack-readout-classifiers-6ab3e85ca7a8152d83c786eb).

References

[1] Ilias Aarab. BTZSC: A benchmark for zero-shot text classification across cross-encoders, embedding models, rerankers and LLMs. In *ICLR*, 2026.

[2] Loubna Ben Allal et al. SmolLM2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

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

[4] Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of EMNLP*, 2020.

[5] Stella Biderman et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

[6] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[7] Tom Brown et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[8] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Workshop on NLP for Conversational AI*, 2020.

[9] Zehua Cheng, Wei Dai, and Jiahao Sun. this-that-model-1.0: A typed decision model that decides in 30 ms, for a millionth of a cent. *arXiv preprint arXiv:2609.23886*, 2026.

[10] Hyung Won Chung et al. Scaling instruction-finetuned language models. *JMLR*, 2024.

[11] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

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

[13] Convai Innovations. Laya. Model card, https://huggingface.co/convaiinnovations/laya, 2026. Apache-2.0 weights released 2026-09-18; accessed 2026-09-21.

[14] Mehul Damani et al. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint arXiv:2507.16806*, 2025.

[15] DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[16] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[17] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *ACL*, 2020.

[18] Jack FitzGerald et al. MASSIVE: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*, 2022.

[19] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *TACL*, 2021.

[20] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

[21] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[22] Franz A. Heinsen. An algorithm for routing vectors in sequences. *arXiv preprint arXiv:2211.11754*, 2022.

[23] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

[24] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *EMNLP*, 2021.

[25] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, 2019.

[26] Edward J Hu et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[27] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *TACL*, 2021.

[28] Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y Fu, Christopher Ré, and Azalia Mirhoseini. Hydragen: High-throughput llm inference with shared prefixes. *arXiv preprint arXiv:2402.05099*, 2024.

[29] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. QASC: A dataset for question answering via sentence composition. In *AAAI*, 2020.

[30] Kimi Team. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[31] Md Kowsher et al. Predicting through generation: Why generation is better for prediction. *arXiv preprint arXiv:2502.17817*, 2025.

[32] Nathan Lambert et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[33] Stefan Larson et al. An evaluation dataset for intent classification and out-of-scope prediction. In *EMNLP*, 2019.

[34] Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. Building efficient universal classifiers with natural language inference. *arXiv preprint arXiv:2312.17543*, 2023.

[35] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *ICLR*, 2025.

[36] Yibo Li et al. Conftuner: Training large language models to express their confidence verbally. *arXiv preprint arXiv:2508.18847*, 2025.

[37] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[38] Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu-lee Wang, Qing Li, and Xiaoqin Zhong. Label supervised llama finetuning. *arXiv preprint arXiv:2310.01208*, 2023.

[39] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *EMNLP*, 2023.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[41] Dongyang Ma, Yan Wang, and Tian Lan. Block-attention for efficient prefilling. In *ICLR*, 2025.

[42] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, 2011.

[43] Meta. Introducing Muse Spark 1.3. https://research.meta.ai/blog/introducing-muse-spark-1-3, 2026.

[44] Meta AI. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024.

[45] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

[46] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *ACL*, 2022.

[47] Mistral AI. Mistral nemo. https://mistral.ai/news/mistral-nemo, 2024.

[48] Minxue Niu and Emily Mower Provost. Contrastive distillation of emotion knowledge from LLMs for zero-shot emotion recognition. *arXiv preprint arXiv:2505.18040*, 2025.

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

[50] James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. I wish i would have loved this one, but i didn't – a multilingual dataset for counterfactual detection in product review. In *EMNLP*, 2021.

[51] OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[52] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

[53] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 2005.

[54] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[55] Qwen Team. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

[56] Nir Ratner et al. Parallel context windows for large language models. In *ACL*, 2023.

[57] Joshua Robinson, Christopher Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*, 2022.

[58] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI*, 2020.

[59] Victor Sanh et al. Multitask prompted training enables zero-shot task generalization. In *ICLR*, 2022.

[60] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In *EMNLP*, 2018.

[61] Ali Şenol, Garima Agrawal, and Huan Liu. Measuring reasoning quality in LLMs: A multi-dimensional behavioral framework. *arXiv preprint arXiv:2605.24661v4*, 2026.

[62] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

[63] Bhavesh Sood and Jaromir Savelka. Punching above their weight: Classification-head fine-tuning of tiny language models (TLMs) for verifiable multiple-choice tasks. *arXiv preprint arXiv:2607.03801*, 2026.

[64] Ihor Stepanov et al. GLiClass: Generalist lightweight model for sequence classification tasks. *arXiv preprint arXiv:2508.07662*, 2025.

[65] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *NAACL*, 2019.

[66] Polina Tsvilodub, Hening Wang, Sharon Grosch, and Michael Franke. Predictions from language models for multiple-choice tasks are not robust under variation of scoring methods. *arXiv preprint arXiv:2403.00998*, 2024.

[67] TypeSafe AI. Introduction – TypeSafe AI. Documentation of Jev, https://docs.typesafe.ai/introduction, 2026.

[68] TypeSafe AI. Introducing system one models & jev. https://typesafe.ai/blog/introducing-system-one-models-and-jev, 2026.

[69] Franklin Wang and Sumanth Hegde. Accelerating direct preference optimization with prefix sharing. *arXiv preprint arXiv:2410.20305*, 2024. URL https://arxiv.org/abs/2410.20305.

[70] Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Röttger, and Barbara Plank. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think. In *COLM*, 2024.

[71] Zhiqiang Wang, Yiran Pang, and Yanbin Lin. Large language models are zero-shot text classifiers. *arXiv preprint arXiv:2312.01044*, 2023.

[72] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

[73] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

[74] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text*, 2017.

[75] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[76] Haike Xu, Zongyu Lin, Jing Zhou, Yanan Zheng, and Zhilin Yang. A universal discriminator for zero-shot generalization. In *ACL*, pages 10559–10575, 2023. doi: 10.18653/v1/2023.acl-long.589. URL https://aclanthology.org/2023.acl-long.589/.

[77] Z.ai. GLM-5.3. https://docs.z.ai/guides/llm/glm-5.3, 2026.

[78] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, 2019.

[79] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

[80] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[81] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *ICML*, 2021.

[82] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *ICLR*, 2024.

[83] Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *ICLR*, 2024.

A Prompts

Every option is rendered as a leading space, the option text and a newline, and the newline is scored. Options are listed in dataset order.

Text classification (`classify_prompt`):

```
<instruction>
Labels: <label 1>, <label 2>, ..., <label n>
Text: <text, cut at 1500 characters>
Label:
```

Banking77, for example, uses the instruction "Classify the customer's banking query into exactly one intent." and the 77 intent names with underscores replaced by spaces.

Multiple choice and yes/no (`mcq_prompt`):

```
<context, if any>
Question: <question>
Options:
- <option 1>
- <option 2>
Answer:
```

Yes/no tasks use the options "yes" and "no".

First-token symbol baseline (`symbol_prompt`): the cue line is replaced by lettered choices "A. <option 1>" …and the letters are scored.

Decision primitives. `choice`: "<state> / Question: <question> / Options: <o1>, …/ Answer:". `score`: levels listed as "<i>: <level>" under "Levels:", ending "Answer (level number):"; the options are the level indices. `true_false`: "Answer (yes or no):" with options yes and no.

Renamed labels (Table 4): SST-5 (very negative to strongly unfavorable, negative to unfavorable, neutral to mixed or indifferent, positive to favorable, very positive to strongly favorable); TweetEval emotion (anger to furious, joy to happiness, optimism to hopeful, sadness to sorrow); TweetEval hate (non-hate to not hateful, hate to hateful); Rotten Tomatoes (negative to thumbs down, positive to thumbs up); SMS Spam (ham to legitimate message, spam to unsolicited junk); Counterfactual (not-counterfactual to describes what actually happened, counterfactual to describes something that did not happen); StrategyQA (yes to true, no to false).

BTZSC. "Pick the label that describes the text." followed by the benchmark's verbalized label descriptions, in the classification template.

B Additional results

Table 11: The 500-example evaluation subsets against the full test sets, Qwen3-1.7B-Base: zero-shot scoring and the trained scalar head (both columns are three-seed means; the full-test runs are three additional runs with the main configuration, reported in Table 3). Differences are at most four points per task (MMLU, TweetEval emotion) and 0.1 on the mean.

task	test size	zero-shot, 500	zero-shot, full	trained, 500	trained, full
banking77	3,076	0.401	0.399	0.680	0.660
massive	2,974	0.673	0.646	0.746	0.740
sst5	2,210	0.341	0.358	0.455	0.451
tweet emotion	1,421	0.698	0.697	0.658	0.694
tweet hate	2,970	0.501	0.484	0.582	0.560
rotten tomatoes	1,066	0.870	0.862	0.903	0.888
sms spam	5,574	0.645	0.676	0.678	0.703
counterfactual	335	0.452	0.451	0.641	0.626
go emotions	4,590	0.194	0.206	0.273	0.295
arc challenge†	1,172	0.734	0.745	0.825	0.838
mmlu†	14,042	0.584	0.552	0.631	0.592
strategyqa	687	0.546	0.548	0.618	0.646
mean		0.553	0.552	0.641	0.641

Table 12: The same check at every size, held-out means: the 500-example subsets against the full test sets, zero-shot and trained (scalar head; at 0.6B one retrained run, at 1.7B three, at 4B and 8B the trained adapter re-evaluated). Means agree within a point except the 0.6B base zero-shot, which the subset understates by 1.2. The one reversal between sizes, the 14B base below the 8B base on MASSIVE, holds on the full test set (0.748 vs. 0.762), so it is not a subset artefact.

size	zero-shot, 500	zero-shot, full	trained, 500	trained, full
0.6B	0.366	0.378	0.568	0.567
1.7B	0.553	0.552	0.641	0.641
4B	0.598	0.594	0.691	0.695
8B	0.627	0.626	0.721	0.721
14B	0.708	0.711	0.714	–

Table 13: Ablations, 1.7B: zero-shot raw and per-byte scoring, first-token symbol scoring, five-shot option scoring (five training examples of the task in the prompt), gold-only SFT, and the trained readouts.

task	zero-shot raw	zero-shot byte	symbol	5-shot	gold-only SFT	scalar head	both heads	set-CE, LM head
banking77	0.401	0.153	–	0.560	0.603	0.680	0.681	0.663
massive	0.673	0.635	–	0.738	0.732	0.746	0.734	0.718
sst5	0.341	0.266	0.232	0.434	0.429	0.455	0.376	0.344
tweet emotion	0.698	0.579	0.646	0.735	0.679	0.658	0.686	0.680
tweet hate	0.501	0.370	0.657	0.391	0.506	0.582	0.539	0.548
rotten tomatoes	0.870	0.870	0.851	0.798	0.900	0.903	0.912	0.903
sms spam	0.645	0.547	0.236	0.749	0.627	0.678	0.607	0.659
counterfactual	0.452	0.453	0.453	0.487	0.483	0.641	0.479	0.569
go emotions	0.194	0.114	–	0.192	0.257	0.273	0.296	0.283
arc challenge†	0.734	0.730	0.716	0.750	0.786	0.825	0.814	0.822
mmlu†	0.584	0.556	0.548	0.590	0.612	0.631	0.632	0.650
strategyqa	0.546	0.597	0.375	0.428	0.642	0.618	0.654	0.633
mean	0.553	0.489	0.524	0.571	0.605	0.641 ± 0.012	0.617	0.623 ± 0.008

Table 14: Accuracy in place of macro-F1 for every held-out task, Qwen3-1.7B-Base, 500-example subsets, columns as in Table 13.

task	zero-shot raw	zero-shot byte	symbol	5-shot	gold-only SFT	scalar head	both heads	set-CE, LM head
banking77	0.432	0.198	–	0.594	0.634	0.707	0.708	0.692
massive	0.656	0.592	–	0.716	0.734	0.767	0.736	0.730
sst5	0.428	0.304	0.292	0.490	0.454	0.472	0.434	0.446
tweet emotion	0.758	0.678	0.734	0.790	0.756	0.729	0.762	0.755
tweet hate	0.612	0.588	0.676	0.460	0.526	0.582	0.544	0.554
rotten tomatoes	0.872	0.872	0.854	0.806	0.900	0.903	0.912	0.903
sms spam	0.796	0.634	0.236	0.844	0.862	0.889	0.720	0.819
counterfactual	0.824	0.827	0.827	0.833	0.824	0.851	0.507	0.677
go emotions	0.232	0.162	–	0.212	0.236	0.327	0.302	0.324
arc challenge	0.734	0.730	0.716	0.750	0.786	0.825	0.814	0.822
mmlu	0.584	0.556	0.548	0.590	0.612	0.631	0.632	0.650
strategyqa	0.582	0.608	0.512	0.542	0.642	0.620	0.654	0.637
mean	0.626	0.562	0.599	0.636	0.664	0.692 ± 0.004	0.644	0.667 ± 0.007

Table 15: Zero-shot held-out accuracy, Qwen3-0.6B-Base, columns as in Table 2.

task	raw	token	byte	char	PMI	+BC	symbol	ECE (raw)
banking77	0.376	0.334	0.258	0.250	0.014	0.436	–	0.073
massive	0.402	0.410	0.396	0.400	0.060	0.494	–	0.143
sst5	0.332	0.268	0.268	0.242	0.124	0.354	0.288	0.368
tweet emotion	0.632	0.632	0.600	0.596	0.636	0.692	0.538	0.163
tweet hate	0.424	0.548	0.500	0.548	0.588	0.602	0.548	0.397
rotten tomatoes	0.602	0.602	0.602	0.602	0.554	0.876	0.604	0.246
sms spam	0.124	0.124	0.122	0.122	0.122	0.514	0.322	0.725
counterfactual	0.675	0.827	0.827	0.827	0.827	0.451	0.663	0.085
go emotions	0.142	0.152	0.106	0.098	0.166	0.200	–	0.369
arc challenge	0.534	0.516	0.516	0.514	0.386	–	0.390	0.137
mmlu	0.414	0.406	0.386	0.384	0.340	–	0.336	0.189
strategyqa	0.512	0.512	0.494	0.496	0.502	0.512	0.498	0.227
mean	0.431	0.444	0.423	0.423	0.360	0.513	0.465	0.260

Table 16: Held-out performance after training, Qwen3-0.6B-Base, 500-example subsets, columns as in Table 3.

task	zero-shot	LM head	scalar head	trained on the task	best prompted frontier (model)	best published zero-shot (size)	best supervised (size)
banking77	0.334	0.611	0.661	–	0.868 (Claude Fable 5.1)	0.715 (8B)	–
massive	0.411	0.686	0.685	–	0.890 (GPT-6 Astra)	0.673 (8B)	–
sst5	0.142	0.281	0.332	–	0.544 (Claude Fable 5.1)	–	–
tweet emotion	0.568	0.652	0.672	–	0.802 (gpt-oss-120b)	0.728 (n.d.)	0.810 (0.12B)
tweet hate	0.314	0.484	0.531	–	0.767 (Kimi K3)	0.708 (20B)	0.581 (0.12B)
rotten tomatoes	0.522	0.873	0.884	–	0.952 (Claude Fable 5.1)	0.903 (8B)	–
sms spam	0.111	0.343	0.455	–	0.986 (Claude Fable 5.1)	–	–
counterfactual	0.515	0.421	0.579	–	0.847 (gpt-oss-120b)	–	–
go emotions	0.112	0.264	0.257	–	0.362 (gpt-oss-120b)	–	–
arc challenge†	0.534	0.666	0.672	–	0.976 (Kimi K3)	0.786 (3.2B)	–
mmlu†	0.414	0.510	0.528	–	0.938 (GPT-6 Astra)	0.626 (1.7B)	–
strategyqa	0.412	0.534	0.560	–	0.923 (Claude Fable 5.1)	–	–
mean	0.366	0.527	0.568				

Table 17: Held-out performance after training, Qwen3-4B-Base, 500-example subsets, columns as in Table 3.

task	zero-shot	LM head	scalar head	trained on the task	best prompted frontier (model)	best published zero-shot (size)	best supervised (size)
banking77	0.578	0.722	0.756	–	0.868 (Claude Fable 5.1)	0.715 (8B)	–
massive	0.728	0.749	0.772	–	0.890 (GPT-6 Astra)	0.673 (8B)	–
sst5	0.393	0.446	0.508	–	0.544 (Claude Fable 5.1)	–	–
tweet emotion	0.710	0.635	0.672	–	0.802 (gpt-oss-120b)	0.728 (n.d.)	0.810 (0.12B)
tweet hate	0.624	0.453	0.566	–	0.767 (Kimi K3)	0.708 (20B)	0.581 (0.12B)
rotten tomatoes	0.887	0.908	0.915	–	0.952 (Claude Fable 5.1)	0.903 (8B)	–
sms spam	0.510	0.592	0.766	–	0.986 (Claude Fable 5.1)	–	–
counterfactual	0.528	0.616	0.764	–	0.847 (gpt-oss-120b)	–	–
go emotions	0.199	0.271	0.282	–	0.362 (gpt-oss-120b)	–	–
arc challenge†	0.836	0.906	0.885	–	0.976 (Kimi K3)	0.786 (3.2B)	–
mmlu†	0.658	0.700	0.702	–	0.938 (GPT-6 Astra)	0.626 (1.7B)	–
strategyqa	0.523	0.708	0.710	–	0.923 (Claude Fable 5.1)	–	–
mean	0.598	0.642	0.691				

Table 18: Held-out performance after training, Qwen3-8B-Base, 500-example subsets, columns as in Table 3.

task	zero-shot	LM head	scalar head	trained on the task	best prompted frontier (model)	best published zero-shot (size)	best supervised (size)
banking77	0.610	0.723	0.755	–	0.868 (Claude Fable 5.1)	0.715 (8B)	–
massive	0.802	0.807	0.809	–	0.890 (GPT-6 Astra)	0.673 (8B)	–
sst5	0.257	0.503	0.526	–	0.544 (Claude Fable 5.1)	–	–
tweet emotion	0.734	0.684	0.661	–	0.802 (gpt-oss-120b)	0.728 (n.d.)	0.810 (0.12B)
tweet hate	0.593	0.536	0.563	–	0.767 (Kimi K3)	0.708 (20B)	0.581 (0.12B)
rotten tomatoes	0.891	0.930	0.927	–	0.952 (Claude Fable 5.1)	0.903 (8B)	–
sms spam	0.692	0.749	0.879	–	0.986 (Claude Fable 5.1)	–	–
counterfactual	0.470	0.805	0.796	–	0.847 (gpt-oss-120b)	–	–
go emotions	0.225	0.299	0.320	–	0.362 (gpt-oss-120b)	–	–
arc challenge†	0.884	0.930	0.927	–	0.976 (Kimi K3)	0.786 (3.2B)	–
mmlu†	0.680	0.756	0.760	–	0.938 (GPT-6 Astra)	0.626 (1.7B)	–
strategyqa	0.685	0.768	0.726	–	0.923 (Claude Fable 5.1)	–	–
mean	0.627	0.707	0.721				

Table 19: Held-out performance after training, gpt-oss-20b (post-trained mixture of experts), 500-example subsets, columns as in Table 3.

task	zero-shot	LM head	scalar head	trained on the task	best prompted frontier (model)	best published zero-shot (size)	best supervised (size)
banking77	0.550	0.808	0.810	–	0.868 (Claude Fable 5.1)	0.715 (8B)	–
massive	0.581	0.764	0.811	–	0.890 (GPT-6 Astra)	0.673 (8B)	–
sst5	0.193	0.332	0.429	–	0.544 (Claude Fable 5.1)	–	–
tweet emotion	0.462	0.650	0.720	–	0.802 (gpt-oss-120b)	0.728 (n.d.)	0.810 (0.12B)
tweet hate	0.631	0.576	0.547	–	0.767 (Kimi K3)	0.708 (20B)	0.581 (0.12B)
rotten tomatoes	0.636	0.902	0.908	–	0.952 (Claude Fable 5.1)	0.903 (8B)	–
sms spam	0.672	0.785	0.765	–	0.986 (Claude Fable 5.1)	–	–
counterfactual	0.500	0.784	0.792	–	0.847 (gpt-oss-120b)	–	–
go emotions	0.092	0.265	0.314	–	0.362 (gpt-oss-120b)	–	–
arc challenge†	0.850	0.900	0.894	–	0.976 (Kimi K3)	0.786 (3.2B)	–
mmlu†	0.614	0.720	0.704	–	0.938 (GPT-6 Astra)	0.626 (1.7B)	–
strategyqa	0.364	0.695	0.675	–	0.923 (Claude Fable 5.1)	–	–
mean	0.512	0.682	0.697				

Table 20: Calibration, 0.6B, pooled over the twelve held-out tasks (5,835 predictions). The zero-shot plateau between 0.5 and 0.9 confidence is the base model confidently wrong on the imbalanced binary tasks (SMS Spam: confidence 0.84, accuracy 0.12; TweetEval hate: 0.82, 0.42); training removes it. Per task, the trained model remains over-confident on TweetEval hate, MMLU and StrategyQA and under-confident on SMS Spam and Amazon Counterfactual (Figure 9).

model	acc	ECE	Brier	NLL	AUROC	acc@50% cov.
zero-shot raw	0.431	0.260	0.754	1.683	0.680	0.550
zero-shot raw + TS	0.431	0.096	0.612	1.339	0.670	0.551
set-CE, LM head	0.556	0.169	0.572	1.213	0.668	0.657
set-CE, LM head + TS	0.556	0.067	0.508	1.035	0.663	0.653
gold-only SFT	0.579	0.153	0.563	1.419	0.693	0.707
set-CE, scalar head	0.634	0.113	0.496	1.017	0.676	0.736

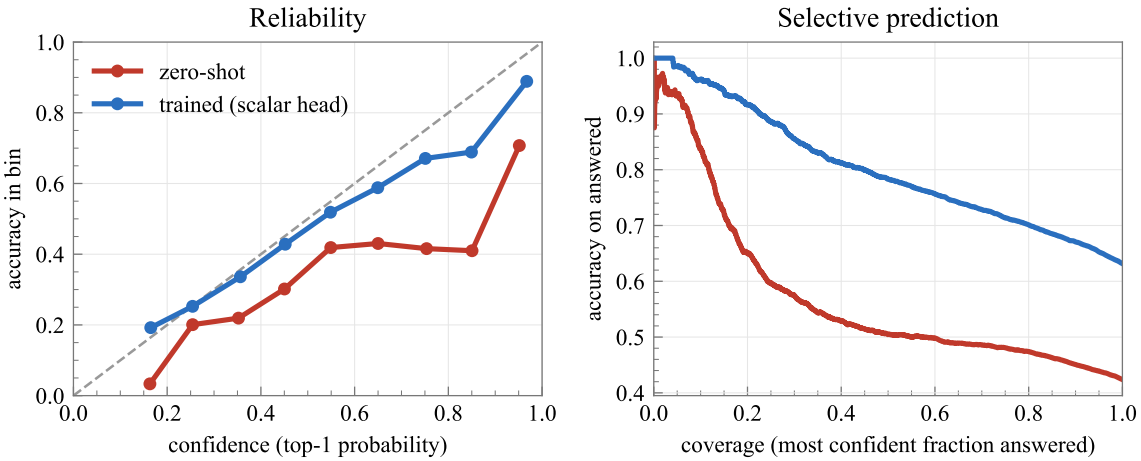


Figure 9: Calibration on held-out tasks, Qwen3-0.6B-Base, zero-shot vs. the trained scalar head, pooled over the twelve tasks as in Figure 4: reliability diagram (left) and accuracy on the most confident fraction (right). The zero-shot plateau between 0.5 and 0.9 confidence is the base model confidently wrong on the imbalanced binary tasks; training removes it (Table 20).

C Supporting tables

These tables supplement the main text: related work, compute per decision, ordinal prediction, prompting failures, adaptation cost, latency, and the label-set and data sweeps.

Table 21: Positioning. "Unseen label sets": evaluated on label sets not trained on. "One pass": all options scored in one forward pass.

method	training	readout	one pass	calibration	unseen label sets	open
Log-likelihood scoring [7]	none	option likelihood	no, one seq./option	poor (Table 2)	yes	yes
First-token symbol [82]	none	P(letter)	yes	letter bias	yes, ≤ 26	yes
T0 / FLAN [59, 72]	mixture, once	option likelihood	no, one seq./option	not measured	yes	yes
Universal Discriminator [76]	mixture, once	encoder scalar head	one seq./option	not measured	yes	yes
Sood et al. [63]	per task	scalar head	no, one seq./option	not measured	no	yes
GLiClass [64]	once	encoder	yes	not stated	yes	yes
NLI / embedding / reranker	once (NLI or retrieval pairs)	pair score	one pass per label	not stated	yes	yes
Jev [67]	undisclosed	distribution	yes	claimed	yes	no
Laya [13]	per domain (RLCD)	encoder mask-token scorer	yes	ECE after temperature	near chance (reported)	yes
Ours, LM head	mixture, once	packed likelihood	yes	proper scores, ranking	yes, 12 tasks	yes
Ours, scalar head	mixture, once	packed $w^\top h$	yes	proper scores, ECE	yes, 12 tasks	yes

Table 22: Size and estimated compute of one decision, ours vs. the systems behind the best published zero-shot numbers. Entries are TFLOPs per decision, $\approx$ 2$\times$ parameters $\times$ tokens processed; token counts are means over our 500 test prompts (Banking77: 365-token prompt, 334 option tokens, 13-token text, 3.5-token labels; SST-5: 51, 12, 23, 1.4). The reranker estimate omits its fixed prompt template, so it is a lower bound. n.d.: size not disclosed.

system	params	one decision	banking77 (77 labels)	sst5 (5 labels)
Ours, Qwen3-1.7B-Base	1.7B	packed scoring: prompt + all options, one pass	2.4	0.2
Qwen3-Embedding-8B	8B	bi-encoder: the text, one pass; labels cached	0.2	0.4
Qwen3-Reranker-8B	8B	cross-encoder: text + label, one pass per label	20.3	1.9
Flan-UL2	20B	generative: the prompt, then decode the label	14.7	2.1
Llama-3.2-3B-Instruct	3.2B	generative: the prompt, then decode the label	2.4	0.3
ChatGPT / GPT-4	n.d.	generative: the prompt, then decode the label	–	–

Table 23: SST-5 (five ordered levels): the argmax of the option distribution against its expected level, rounded (each cell: argmax / expected). Mean absolute error in levels, share of predictions off by two or more levels, and Spearman correlation with the gold level. The expected level is the better ordinal prediction for every model, by the widest margin for the trained 1.7B scalar head.

model	accuracy	MAE (levels)	off by ≥ 2 levels	Spearman ρ
1.7B zero-shot	0.428	0.666 / 0.642	0.074 / 0.054	0.751 / 0.813
1.7B LM head	0.446	0.623 / 0.608	0.064 / 0.056	0.765 / 0.829
1.7B scalar head	0.472	0.636 / 0.549	0.087 / 0.055	0.759 / 0.831
8B zero-shot	0.286	0.982 / 0.862	0.230 / 0.192	0.774 / 0.847
8B scalar head	0.535	0.528 / 0.484	0.059 / 0.034	0.819 / 0.862

Table 24: A trained single-token readout against pack-readout, Qwen3-1.7B-Base, seed 0, on the same 500 test examples per task (the first 500 of each shuffled test set; pack-readout: the seed-0 full-test adapters of Table 3). Top: means over the twelve held-out tasks; bottom: primary score per task. Single-token: the option letters' logits at one answer position, trained with the same LoRA, objective and mixture.

readout	mean primary	accuracy	Brier	NLL	ECE
single-token	0.626	0.679	0.447	1.045	0.116
pack-readout, scalar head	0.652	0.696	0.414	0.875	0.097
pack-readout, LM head	0.617	0.650	0.466	1.036	0.120

task	n	single-token	pack-readout, scalar head	pack-readout, LM head
banking77	500	0.631	0.666	0.649
massive	500	0.691	0.732	0.728
sst5	500	0.453	0.488	0.388
tweet emotion	500	0.692	0.706	0.688
tweet hate	500	0.583	0.554	0.528
rotten tomatoes	500	0.902	0.912	0.898
sms spam	500	0.664	0.716	0.655
counterfactual	335	0.594	0.650	0.497
go emotions	500	0.247	0.265	0.291
arc challenge†	500	0.794	0.824	0.808
mmlu†	500	0.598	0.642	0.626
strategyqa	500	0.660	0.668	0.654

Table 25: Invalid replies of the prompted Qwen3-1.7B: the first two per task that named no option, with the gold label. Each is a plausible answer outside the option set, which option scoring cannot produce.

task	reply	gold option
banking77	"reverted top up"	top up reverted
banking77	"freeze my account"	compromised card
massive	"lists addorcreate"	calendar set
massive	"recommendations movies"	recommendation movies
tweet emotion	"angry"	anger
tweet emotion	"sad"	sadness
rotten tomatoes	"neutral"	positive
rotten tomatoes	"more intellectually scary than dramatically involving"	negative
sms spam	"A"	spam
go emotions	"desperation"	neutral
go emotions	"redeem"	neutral
arc challenge	"the color of the light bulb they use"	the color of light bulb they use
arc challenge	"release of chlorofluorocarcs from aerosol cans"	release of chlorofluorocarbons from aero
mmlu	"18x + 15"	18x+15
mmlu	"recognized as breaking an important norm of behaviour"	seen a temporary aberration from an othe

Table 26: Adaptation data and compute beyond pretraining, where disclosed. Ours: 16,000 public examples, 20 minutes on one A100 at 1.7B.

system	adaptation data	compute	data public
Ours, from Qwen3-1.7B-Base	16,000 examples from 18 public datasets, 2.5M tokens	20 min, one A100 (25 PFLOPs)	yes
Ours, from Qwen3-8B-Base	same	2.1 h, one DGX Spark (120 PFLOPs)	yes
gte-large-en-v1.5 [37]	MLM on C4; 800M web-mined pairs; 3M annotated pairs	n.d. (v1: 50k pre-training steps at batch 16k on 8 A100-80GB)	sources only
Qwen3-Embedding-8B [80]	150M synthetic pairs; 7M labeled + 12M synthetic pairs	n.d.	sources only
Qwen3-Reranker-8B [80]	7M labeled + 12M synthetic pairs	n.d.	sources only
deberta-v3-large-nli-triplet [1]	885k NLI pairs, 3 epochs [34]	n.d. (a similar run: 10 h, one A100)	yes
Flan-T5-XXL [10]	Flan 2022 collection, 1,836 tasks	7.6×10^{19} FLOPs (76,000 PFLOPs)	yes
Mistral-Nemo-Instruct [47]	n.d.	n.d.	no
Llama-3.2-3B-Instruct [44]	SFT, rejection sampling, DPO; counts n.d.	460k H100 hours with pre-training	no
GPT-5, Claude	n.d.	n.d.	no

Table 27: Latency per decision (median ms over 20 inputs and three random label sets per count), Qwen3-1.7B-Base, bf16, A100-40GB, labels listed in the prompt.

options	prompt tok.	packed (ours)	KV replication	one seq./option	constrained decoding	JSON generation
2	37	47.5	91.4	47.3	203.3	1042.8
4	44	47.3	89.9	48.2	163.6	1030.7
8	57	47.3	90.4	49.2	203.2	1014.1
16	83	48.2	92.3	50.8	164.8	1033.8
77	295	54.4	106.9	578.0	167.9	1042.8
255	921	90.8	432.8	5606.1	217.2	1184.6

Table 28: Held-out performance vs. number of training label sets at 16,000 example presentations and 2,000 steps, scalar head, Qwen3-1.7B-Base: mean and s.d. over three random subsets of the training tasks per count (at 18, over three training seeds).

training label sets	held-out acc	many-label acc	ECE	AUROC
4	0.638 ± 0.014	0.471	0.149	0.718
6	0.663 ± 0.020	0.562	0.142	0.724
8	0.655 ± 0.025	0.568	0.130	0.731
10	0.669 ± 0.025	0.574	0.109	0.733
12	0.667 ± 0.004	0.563	0.125	0.730
16	0.672 ± 0.011	0.589	0.120	0.738
18	0.693 ± 0.003	0.607	0.095	0.745

Table 29: Held-out performance vs. the size of the training pool at 2,000 steps, scalar head, Qwen3-1.7B-Base; each of the 18 tasks is capped at the per-task count, and the smallest pool is seen eight times.

examples in the pool	per task	held-out macro-F1	many-label acc	ECE
1,998	111	0.595	0.570	0.311
7,992	444	0.623	0.607	0.195
16,002	889	0.631	0.598	0.115
36,000	2,000	0.641	0.600	0.098

D BTZSC comparison by task group

Table 30 gives the numbers behind Figure 8 (Section 5.8), overall and per task group.

Table 30: Comparison with dedicated zero-shot classifiers on the BTZSC benchmark (22 datasets; macro-F1 overall and per task group), under the benchmark's own protocol: every system receives the text and a verbalized description of each label, and none is trained on the task. Top: the best model of each family in the leaderboard and its small LLMs, on full test sets. Bottom: ours, on a seeded 500-example subset per dataset; the trained rows use the training mixture without the five datasets that BTZSC also contains. Compared with 8B retrieval models and 4–12B LLMs, our model is within 2 points of the best system at 1.7B, exceeds it at 4B, half its size, and exceeds it by two points at 8B, the same size.

model	overall	sentiment	topic	intent	emotion
Qwen3-Reranker-8B (reranker, 8.0B)	0.722	0.923	0.662	0.700	0.486
Mistral-Nemo-Instruct-2407 (LLM, first token, 12.2B)	0.670	0.841	0.691	0.455	0.363
Qwen3-4B (LLM, first token, 4.0B)	0.649	0.883	0.638	0.401	0.373
gte-large-en-v1.5 (embedding, 0.4B)	0.617	0.849	0.543	0.590	0.372
deberta-v3-large-nli-triplet (NLI, 0.4B)	0.596	0.899	0.503	0.447	0.419
Llama-3.2-3B-Instruct (LLM, first token, 3.2B)	0.430	0.456	0.437	0.408	0.348
Ours, Qwen3-1.7B-Base, zero-shot (no training)	0.526	0.726	0.489	0.352	0.386
Ours, 1.7B trained, LM head (mixture without BTZSC data)	0.672	0.913	0.598	0.605	0.456
Ours, 1.7B trained, scalar head (mixture without BTZSC data) (2 seeds)	0.703	0.899	0.648	0.658	0.480
Ours, 0.6B trained, scalar head (same)	0.673	0.879	0.621	0.613	0.433
Ours, 4B trained, scalar head (same) (2 seeds)	0.727	0.929	0.669	0.721	0.444
Ours, 8B trained, scalar head (same) (2 seeds)	0.740	0.929	0.691	0.731	0.454
Ours, 14B trained, scalar head (same; 1,000 steps, half the learning rate)	0.745	0.932	0.689	0.760	0.473

E Published numbers on the held-out tasks

Best published results we could verify (2026-09-19, corrected 2026-09-30) per held-out task, with model size and the metric each is reported in. Zero-shot means no training on the dataset. Sources: BTZSC results (Aarab [1], full test sets); Zhang et al. 2023 (arXiv 2305.15005; 500 sampled test items); TimeLMs (arXiv 2202.03829); Casanueva et al. 2020 (arXiv 2003.04807); FitzGerald et al. 2022 (arXiv 2204.08582); Emo Pillars (arXiv 2504.16856); the Llama 3.2 and Qwen3 model reports; for SMS Spam, Wang et al. [71] (150 sampled test messages) and Houlsby et al. [25] (full fine-tuning; their 558-message test split); for Counterfactual, Lee et al. [35] (a logistic probe on frozen embeddings; MTEB's English test set); for GoEmotions, Niu and Mower Provost [48]; for SST-5, Heinsen [22]; for StrategyQA, Şenol et al. [61] (version 4; 250 sampled items of the test split we use). The published supervised StrategyQA results use other splits. GoEmotions numbers are multi-label and not comparable with our single-label subset.

Table 31: Published numbers on the held-out tasks.

task	best zero-shot (model, metric)	best supervised (model, metric)
banking77	0.715 Qwen3-Embedding (8B) (macro-F1)	0.937 BERT-large (0.3B) (accuracy)
massive	0.673 Qwen3-Reranker (8B) (macro-F1)	0.890 mT5-base (0.6B) (accuracy)
sst5	0.570 Flan-UL2 (20B) (accuracy)	0.598 routing head on frozen RoBERTa-large (0.4B) (accuracy)
tweet emotion	0.728 ChatGPT (macro-F1)	0.810 TimeLM (0.1B) (macro-F1)
tweet hate	0.708 Flan-UL2 (20B) (macro-F1)	0.581 TimeLM (0.1B) (macro-F1)
rotten tomatoes	0.903 Qwen3-Reranker (8B) (macro-F1)	0.900 T5-large (0.8B) (accuracy)
sms spam	0.973 GPT-4 (accuracy)	0.993 BERT-base (0.1B) (accuracy)
counterfactual	none found	0.951 NV-Embed-v1 (8B), probe on 32 examples per label (accuracy)
go emotions	0.319 GPT-4 (macro-F1, multi-label)	0.550 RoBERTa-large (0.4B) (macro-F1, multi-label)
arc challenge	0.786 Llama-3.2-3B-Instruct (accuracy)	none found
mmlu	0.626 Qwen3-1.7B-Base, 5-shot (accuracy)	none found
strategyqa	0.524 Phi-2 (2.7B), free-form answer (accuracy)	none found

F Prompting baselines: protocol

Every prompted model receives the scorer's prompt without its cue line, followed by "Reply with one of the listed answers, exactly as written, and nothing else.", as a single user message in the model's chat format; thinking is disabled where the template allows it. Local models sample free replies with the settings their authors recommend for the non-thinking mode (temperature 0.7, top-p 0.8, top-k 20; seed 0) and decode constrained replies greedily. Hosted requests set temperature 0; the gateway does not report whether a provider applies it to a reasoning model. A reply is matched to an option by exact match after normalization (case, surrounding quotes and punctuation, whitespace), else by the longest option contained in the reply as whole words, else by the single option the reply is part of; otherwise it is invalid and counts as wrong. Every reply is stored with the results. Local models (Qwen3-1.7B, 4B, 8B, the post-trained releases) run on a Lambda A100-40GB one decision at a time; free replies end at the end-of-turn token or at a 32,768-token cap, which two MMLU replies of the 1.7B model reached. Hosted models run through the Vercel AI Gateway. Muse Spark 1.3 and GLM-5.3 have no request-level output-token cap; the other hosted runs used a 1,024-token budget including hidden reasoning. Requests are restricted to the providers listed in Table 32. Serving providers are recorded per task, and per reply for Muse Spark 1.3 and GLM-5.3. GPT-5, GPT-6 Astra and Muse Spark 1.3 run with minimal reasoning effort, GLM-5.3 with low effort; the other models use their default settings, which for Claude Sonnet 5 and Claude Fable 5.1 means adaptive thinking on and for Claude Sonnet 4.6 no thinking (the labels of Table 7 state the effective setting). Table 32 lists the model ids, the providers that served the requests and dates; Appendix I lists available technical reports.

G Reproduction

The released repository contains the model implementation, training-data loaders, and training and evaluation entry points. `scripts/train.py` trains the LoRA adapter and readout; `scripts/evaluate.py` evaluates a base model or a saved adapter on the specified tasks; `scripts/prompting_baseline.py`, `scripts/prompting_baseline_api.py`, `scripts/bench_latency.py` and `scripts/btzsc_eval.py` run the prompting, latency and BTZSC comparisons of Sections 5.5, 5.6 and 5.8. Section 4 gives the training settings. The results archive (Data availability) holds the evaluation reports behind every reported number. The trained adapters (LoRA weights and, for the scalar readout, the head vector) of the main runs at every size, the other base models, the LM-head readout, the BTZSC-clean mixture and the additional seeds are released on Hugging Face, named by base model (`https://huggingface.co/collections/maxz411/pack-readout-classifiers-6ab3e85ca7a8152d83c786eb`), under CC BY-NC 4.0 because the training mixture includes SciQ (CC BY-NC 3.0).

Table 32: Hosted models of Table 7: gateway model id, total parameters as stated by the model's developer (n.d.: not disclosed), serving providers, request settings and date of the run.

model id	params (B)	served by	settings	date
alibaba/qwen-3-235b	235	deepinfra, novita	default	2026-09-19
anthropic/claude-fable-5.1	n.d.	anthropic	default	2026-09-20
anthropic/claude-sonnet-4.6	n.d.	anthropic	default	2026-09-19
anthropic/claude-sonnet-5	n.d.	anthropic	default	2026-09-20
deepseek/deepseek-v3.2	671	deepinfra	default	2026-09-19
meta/muse-spark-1.3	n.d.	meta	reasoning_effort=minimal	2026-09-20
moonshotai/kimi-k3	2,800	alibaba, baseten	default	2026-09-20
openai/gpt-5	n.d.	openai	reasoning_effort=minimal	2026-09-19
openai/gpt-6-astra	n.d.	openai	reasoning_effort=minimal	2026-09-20
openai/gpt-oss-120b	117	baseten	default	2026-09-19
zai/glm-5.3	753	deepinfra	reasoning_effort=low	2026-09-20

One training run diverged: the seed-2 full-test run of the 1.7B scalar head on the Lambda A100s (its loss rose from about 0.5 to 1.3 after step 500 and stayed there, and its held-out scores were near chance). It was rerun with the same seed; only the rerun is reported. A concurrent seed-3 run reached a held-out mean of 0.649 and is not used. No other run diverged. At 4B, the scalar-head seed 0 (`4b-set-scalar-s0`) was an early 1,000-step run and seed 1 the 2,000-step recipe; the two agree within 0.005 on the held-out mean, and the 1,000-step control of Section 5.7 shows that the step count does not change the result at 4B.

H Latency at other sizes

The sweep of Table 27 at 0.6B and 4B, same protocol and GPU. At 4B, KV replication does not fit in 40 GB at 255 options.

Table 33: Latency per decision (median ms), Qwen3-0.6B-Base, A100-40GB, columns as in Table 27.

options	prompt tok.	packed (ours)	KV replication	one seq./option	constrained decoding	JSON generation
2	37	44.7	85.4	45.7	193.9	798.3
4	44	45.3	87.3	46.3	161.7	778.8
8	57	45.2	87.2	46.5	195.6	860.4
16	83	46.1	87.7	48.4	158.1	850.6
77	295	52.9	99.5	340.8	162.9	931.9
255	921	72.6	406.6	3291.1	173.0	969.7

Table 34: Latency per decision (median ms), Qwen3-4B-Base, A100-40GB, columns as in Table 27; OOM: does not fit in 40 GB.

options	prompt tok.	packed (ours)	KV replication	one seq./option	constrained decoding	JSON generation
2	37	59.8	116.6	61.7	260.7	1351.5
4	44	59.9	114.8	61.2	212.1	1355.3
8	57	61.2	116.4	62.6	209.3	1364.6
16	83	62.4	119.6	89.7	210.9	1385.4
77	295	69.2	163.9	1237.8	219.4	1373.7
255	921	159.0	OOM	12349.9	234.5	1433.0

I Sources of datasets and models

Training mixture: AG News, DBpedia-14, Yahoo Answers and Yelp-5 [79]; IMDB [42]; TweetEval sentiment, offensive and irony [4]; Emotion [60]; CLINC-150 [33]; SciQ [74]; OpenBookQA [45]; CommonsenseQA [65]; QASC [29]; ARC-Easy [12]; HellaSwag [78]; WinoGrande [58]; BoolQ [11]. Held out: Banking77 [8]; MASSIVE [18]; SST-5 [62]; Rotten Tomatoes [53]; TweetEval emotion and hate [4]; GoEmotions [17]; SMS Spam [3]; Amazon Counterfactual [50]; ARC-Challenge [12]; MMLU [23]; StrategyQA [19]. Base models: Qwen3 [55], Qwen2.5 [54], OLMo 2 [49], SmolLM2 [2]. Hosted open-weight models with a published report: gpt-oss [51], DeepSeek-V3 [15], the line that V3.2 continues, Kimi K2 [30], the predecessor of K3, and Qwen3-235B [55]. GLM-5.3 [77] and Muse Spark 1.3 [43] are documented by their developers; the closed models are identified by model id and date in Table 32. BTZSC reference systems of Table 30: Qwen3-Reranker-8B [80], gte-large-en-v1.5 [37], Llama-3.2-3B-Instruct [44] and Mistral-Nemo-Instruct [47].