Fine-Tuning Causal LLMs for Text Classification: Embedding-Based vs. Instruction-Based Approaches

Anonymous ACL submission

Abstract

We explore efficient strategies to fine-tune decoder-only Large Language Models (LLMs) for downstream text classification under resource constraints. Two approaches are investigated: (1) attaching a classification head to a pre-trained causal LLM and fine-tuning on the task (using the LLM's final token embedding as a sequence representation), and (2) instruction-tuning the LLM in a *prompt→response* format for classification. To enable single-GPU fine-tuning of models up to 8B parameters, we combine 4-bit model quantization with Low-Rank Adaptation (LoRA) for parameter-efficient training. Experiments on two patent benchmarks — a 5-class single-label internal corpus and the public WIPO-Alpha multi-label dataset (14 categories) — show that the embedding-head approach matches or exceeds fine-tuned BERT baselines on single-label classification while training 10–30× fewer parameters; instruction-tuning is competitive only in the multi-label regime, and only with substantially larger trainable budgets (≥100M parameters). These results demonstrate that directly leveraging the internal representations of causal LLMs, along with efficient fine-tuning techniques, yields impressive classification performance under limited computational resources. We discuss the advantages of each approach while outlining practical guidelines and future directions for optimizing LLM fine-tuning in classification scenarios.

1 Introduction

Large Language Models (LLMs) with billions of parameters have recently shown strong natural language generation and understanding capabilities (Brown et al., 2020; Wei et al., 2022; Ji et al., 2023; Chowdhery et al., 2023). Traditionally, text classification is performed by fine-tuning encoder-based transformers such as BERT or RoBERTa on labeled data (Devlin et al., 2019; Liu et al., 2019), using a special classification token (e.g., [CLS]) whose final hidden state feeds a linear classifier. In contrast, decoder-only (causal) LLMs are pre-trained for next-word prediction with a left-to-right attention mask, so tokens only attend to preceding context and there is no explicit classification token or single bidirectional sequence representation, despite the models being trained at scale on vast corpora. This raises the question: can we effectively fine-tune large causal LLMs for classification, and do they outperform traditional domain-specific models on downstream tasks?

Fine-tuning such LLMs is challenging due to their size (Ding et al., 2023; Chen et al., 2024), as full-parameter updates for multi-billion-parameter models are often infeasible on a single GPU. Recent parameter-efficient fine-tuning (PEFT) methods address this (Ding et al., 2023; Fu et al., 2023; Lialin et al., 2023). Low-Rank Adaptation (LoRA) freezes the original weights and injects small trainable matrices in each layer (Hu et al., 2021), drastically reducing trainable parameters while largely preserving performance, including on GPT-3-scale models (Hu et al., 2021). Complementary quantization techniques, such as 4-bit loading via the BitsAndBytes library,[1] substantially cut memory with minimal accuracy loss (Zheng et al., 2024), and QLoRA (Dettmers et al., 2023) combines 4-bit quantization with LoRA-based fine-tuning to enable single-GPU adaptation of models up to 65B parameters.

In this work, we leverage these innovations to fine-tune several open-source causal LLMs—including LLaMA variants (Touvron et al., 2023) (1B, 3B, 8B), and recent models such as Mistral-7B (Jiang et al., 2023), Qwen (Bai et al., 2023), Gemma (Team et al., 2024), and Microsoft's Phi (Abdin et al., 2024)—for text classification in the patent domain. We compare two methodolo-

[1] https://pypi.org/project/bitsandbytes/

gies: (1) an embedding-based approach that adds a classification head on top of the final-token embedding, treating the LLM as a feature extractor analogous to the [CLS] representation in BERT, and (2) an instruction-based approach that reformulates classification as an instruction-following generation problem (Wei et al., 2021; Sanh et al., 2021). In Approach 1 a lightweight feed-forward head is trained on the final-token embedding, while in Approach 2 the model is fine-tuned to generate the correct label given a prompt describing the task.

To our knowledge, this is the first systematic comparison of (i) embedding-head adaptation and (ii) instruction-tuning of decoder-only LLMs as classifiers under a strict 24 GB single-GPU constraint, across 20 modern LLM families and two patent-classification tasks. Our embedding-head method is the supervised, single-task analogue of LLM2Vec step 0 (BehnamGhader et al., 2024) – frozen-base, last-token-pooled, with a discriminative head trained on labeled data – without contrastive pre-training or bidirectional-attention conversion. Our contributions are:

- *Head-to-head empirical comparison* backed by paired McNemar tests and bootstrap ΔF1 95% confidence intervals (Appendix J), F1 and throughput results across an order-of-magnitude trainable-parameter range, and external validation on AG News (Appendix K).

- *Regime characterization:* Embedding-head adaptation matches or exceeds instruction tuning and BERT baselines on single-label classification with $\leq$25M trainable parameters; instruction-tuning is competitive on multi-label, but requires $\geq$100M trainable parameters to do so – a caveat to the "parameter-efficient" framing in this regime.

- *Mechanistic ablations* that explain *why* the embedding head wins under our constraints: a pooling-strategy ablation (Appendix M), calibration measurements (ECE/Brier; Appendix N), and a verbalizer-design ablation isolating prompt-template brittleness in the instruction approach (Appendix P).

- *Practitioner recipes:* Implementation recommendations for fine-tuning causal LLMs as classifiers on a single 24 GB GPU, plus a knowledge-distillation recipe (Appendix O) that recovers BERT-class throughput at a $\leq$1.5-point F1 cost — closing the deployability gap to ModernBERT.

Code, configurations, and prompt templates will be released upon acceptance.[2]

2 Related Work

Causal LLMs as text encoders. A growing line of work treats decoder-only LLMs as feature extractors for downstream discriminative tasks. BehnamGhader et al. (2024) introduce LLM2Vec, an unsupervised three-step recipe (bidirectional attention, masked next-token prediction, contrastive learning) that converts any causal LLM into a state-of-the-art text encoder on MTEB. Earlier supervised variants include adapting T5-style encoders for classification (Wei et al., 2021) and few-shot sentence-transformer methods such as SetFit (Tunstall et al., 2022). Our Approach 1 occupies the supervised, single-task, frozen-base region of this design space: we attach a discriminative head to the last-token hidden state, with LoRA adapters injected into the base, without contrastive pretraining or bidirectional-attention conversion.

Parameter-efficient adaptation. Parameter-efficient fine-tuning (PEFT) methods (Ding et al., 2023; Fu et al., 2023; Lialin et al., 2023) reduce trainable-parameter cost while preserving accuracy. Beyond LoRA (Hu et al., 2021, 2022) and its quantized variant QLoRA (Dettmers et al., 2023), prompt-based families—prompt tuning (Lester et al., 2021), P-tuning v2 (Liu et al., 2022)—provide an alternative axis of efficiency. We pair LoRA with 4-bit NF4 quantization for both approaches, fixing this design choice to isolate the embedding-vs-instruction contrast.

Instruction tuning and classification. FLAN and T0 (Wei et al., 2021; Sanh et al., 2021) demonstrate that multi-task instruction tuning yields strong zero- and few-shot classifiers, and concurrent work continues to evaluate instruction-tuned decoders on classification tasks (Wang et al., 2024; Fatemi et al., 2024; Yin et al., 2024). Ghosh et al. (2024) document calibration and stability issues that motivate caution. For patent classification specifically, encoder-based baselines include PatentBERT and SBERT-based variants such as PatentSBERTa (Bekamiri et al., 2021); we compare against PatentSBERTa directly on WIPO (Ap-

[2] Anonymous artifact link will be provided for the rebuttal phase.

pendix L) and against four BERT-class baselines (Anonymous, 2025) in the main results table.

3 Methodology

We describe the two fine-tuning approaches in detail, along with the techniques used to make training feasible on limited hardware. Specifically, all experiments were run on a **Databricks** cluster using runtime 16.4 LTS (Apache Spark 3.5.2, Scala 2.12), on a **g6.2xlarge** node equipped with a single **NVIDIA L4** GPU and **32 GB** host memory. Aggressive quantization and PEFT techniques were essential to make training feasible under these constraints.

3.1 Approach 1: Classification Head on LLM Embeddings

Our first approach treats the causal LLM as an encoder that produces a fixed-size representation of the input, which we then feed to a small classification head. Specifically, given an input text sequence, we append the end-of-sequence token (if not already present) and obtain the hidden state of the final token from the LLM's last layer. This final token's embedding serves as a holistic representation of the entire sequence (analogous to BERT's [CLS] embedding which captures an "aggregate" of the input). We attach a simple linear layer (or two-layer feed-forward network) on top of this embedding to predict the class label. Because causal LLMs process input autoregressively, the last token's state inherently attends to all previous tokens in the sequence. Thus, using that state as a summary vector allows the classifier to consider the full context of the input. We chose last-token over mean-, max-, and a learnable `[CLS]`-pooled alternative; the ablation in Appendix M (Table 12) confirms last-token is within $\leq$0.01 F1 of mean-pool on both datasets and substantially outperforms max-pool and learnable `[CLS]` (which collapses on multi-label).

During fine-tuning, only the classification head and LoRA adapters ψ injected into the query/key/-value/output projections and feed-forward layers are updated; the base weights θ remain frozen and are loaded in 4-bit NF4 precision via `BitsAndBytes`, compressing memory to roughly $\frac{1}{8}$ of bfloat16 and fitting 8B-parameter models on a 24 GB GPU (QLoRA; Dettmers et al., 2023). We sweep LoRA rank $r \in \{8, 16\}$ with $\alpha = 16$ and 5% adapter dropout, in line with the original LoRA formulation and standard PEFT practice (Hu et al., 2021, 2022; Raschka, 2023; Unsloth, 2025).

Multi-label. For multi-label tasks the head emits per-class sigmoids trained with BCE (Eq. 2) and thresholded at 0.5 at inference.

Notation. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a labeled dataset. For single-label classification, $y_i \in \{1, \ldots, C\}$; for multi-label classification, $y_i \in \{0, 1\}^C$. A causal LLM with frozen base parameters θ and trainable LoRA adapters ψ maps a tokenized input x of length T to hidden states $H = f_{\theta,\psi}(x) \in \mathbb{R}^{T \times d}$. We denote the last-token state by $s(x) \coloneqq H_T \in \mathbb{R}^d$. Unless otherwise noted, gradients update only ψ and any task-specific trainable layers, while θ remains frozen.

We attach a light-weight classifier to the last-token embedding $s(x)$. Let $\Theta_{\text{head}} = \{W, b\}$ with $W \in \mathbb{R}^{C \times d}$ and $b \in \mathbb{R}^C$, and define logits $z(x) \coloneqq Ws(x) + b$.

Objectives. For single-label classification, with $p(c \mid x) = \mathrm{softmax}(z(x))_c$, we minimize the cross-entropy

$$\mathcal{L}_{\text{cls}}(\psi, \Theta_{\text{head}}) = -\tfrac{1}{N} \sum_i \log p(y_i \mid x_i). \quad (1)$$

For multi-label, with $\sigma(\cdot)$ the elementwise logistic and $y_i \in \{0, 1\}^C$, we minimize binary cross-entropy

$$\mathcal{L}_{\text{ml}}(\psi, \Theta_{\text{head}}) = -\tfrac{1}{N} \sum_{i,c} \big[y_{ic} \log \sigma(z_c) + (1 - y_{ic}) \log(1 - \sigma(z_c)) \big], \quad (2)$$

and threshold $\sigma(z_c(x))$ at 0.5 at inference. Both losses are minimized over ψ and Θ_{head}, with θ frozen.

3.2 Approach 2: Instruction-Tuning for Classification

Our second approach frames the classification problem as a form of natural language instruction following recent research (Wang et al., 2024; Fatemi et al., 2024; Yin et al., 2024). Instead of extracting embeddings, we convert each training example into a prompt for the LLM, and the label into a target response. For example, given an input text and a label, we might construct a prompt like: *"Text: <input text>\nQuestion: What is the category of the above text?\nAnswer:"* and train the LLM to output the correct category in the answer portion. The

model is fine-tuned on many such *prompt→answer* pairs so that it learns to produce the appropriate label when given a query prompt.

We fine-tune the LLM on these *prompt→answer* pairs using supervised learning (standard next-token prediction loss on the label tokens). Similar to Approach 1, we apply 4-bit quantization and LoRA adapters (r=64) to reduce memory usage. LoRA target layers are the same as described above, and we again freeze the main weights and train only the LoRA parameters and final linear layers.[3]

Single-label prompting. For our LLM classifier, we map each canonical label to a short, stable identifier, using capital letters `A`, `B`, `C`, ... (e.g., `A: novelty_high`, `B: novelty_medium`, etc.). At inference time, the model receives (i) an instruction to choose exactly one label, (ii) a space-separated list of identifier–label pairs, and (iii) the input text. The model is instructed to answer in a strict format `<ID>\t<LABEL_NAME>` and to output nothing else. During training, we also include the gold identifier and label name after the `ANSWER:` prefix. The full prompt templates are given in Appendix E.

Multi-label prompting. For multi-label classification tasks, we provide the model with an allowed set of labels and instruct it to choose zero or more labels. The model is required to return a machine-readable answer in the form of a JSON-style list under a single key `labels`, for example `labels: ["label_1", "label_3"]` or `labels: []`. During training, we include the gold label set after the `labels:` prefix; during testing, the value is left empty for the model to complete. The exact prompt templates are given in Appendix E.

Prompting and verbalization. Let $\mathcal{V}$ be the tokenizer vocabulary and τ : text $\to \mathcal{V}^*$ the tokenizer. We write $p(x)$ for a prompt-construction function that renders the instruction plus input x (e.g., lists the label choices and ends with `ANSWER:`). A *verbalizer* is a mapping $V : \{1, \ldots, C\} \to$ text that assigns a short, canonical string to each class (e.g., an ID–name pair such as `A\tEdge Security`). For multi-label cases, V extends to $V : \{0, 1\}^C \to$ text by concatenating the positive label verbalizations with a delimiter. We denote the token sequence of the (single- or multi-label) verbalization by $\ell(y) := \tau\big(V(y)\big) = (\ell_1, \ldots, \ell_L)$.

Each example is rendered as a prompt $p(x)$ (instruction + input). The gold label is verbalized and tokenized as $\ell(y) = (\ell_1, \ldots, \ell_L)$. Training uses next-token prediction (teacher forcing) over the label tokens.[4]

Objective. Under autoregressive factorization, the loss is the token-level NLL over the label verbalization:

$$\mathcal{L}_{\text{inst}}(\psi) = -\tfrac{1}{N} \sum_{i,t} \log p_{\theta,\psi}\big(\ell_{i,t} \mid p(x_i), \ell_{i,1:t-1}\big), \tag{3}$$

where for multi-label $\ell(y_i)$ is the delimiter-joined list of positive-label verbalizations and the loss form is unchanged; we mask the prompt (answer-only loss) in all experiments. At inference, decoding is constrained to $\{\ell(1), \ldots, \ell(C)\}$ and terminated at a delimiter or end token. Approach 1 therefore optimizes calibrated C-way class posteriors (Eqs. 1–2) while Approach 2 optimizes token-level likelihood (Eq. 3) and requires constrained decoding; both freeze θ and update ψ, with full hyperparameters in Table 5 (Appendix C).

4 Results and Observations

We evaluated our fine-tuning approaches on two text classification datasets from the patent domain (see Appendix D for detailed dataset statistics and label hierarchies). In particular, we report results on a proprietary internal dataset for single-label classification (described in Appendix D) and the public WIPO patent dataset[5] (a multi-label hierarchical classification task). We compare the proposed methods with a baseline of smaller BERT-based classifiers. Our main evaluation metric is F1-score (with micro-averaging for class-imbalanced scenarios).

4.1 Direct Comparison of Embedding-Based and Instruction-Based Methods

Table 1 compares Approaches 1 and 2 on five LLMs spanning the 1B–7B regime (Gemma-2-2B, Llama-3.2-1B / 1B-Instruct / 3B, Mistral-7B-v0.3), chosen to cover three open-source families and to anchor the rank-comparison sweep ($r \in \{8, 16\}$); the broader 20-model evaluation appears in Table 4. Two patterns dominate the table. *First*, Approach 1

[3] Single exception: Gemma-3-270m fits a 24 GB GPU under full fine-tuning at 270M parameters, so we train all of its parameters directly rather than via LoRA; see Appendix C.

[4] In practice one can either mask the prompt tokens (*answer-only* loss) or include them in the loss (*full-sequence* loss).

[5] https://www.wipo.int/portal/en/index.html

Appr.	Model	Tot.	Train.	INT	WIPO
A1 (r=8)	Gemma-2-2B	2.6B	10.4M	0.832	0.779
	Llama-3.2-1B	1.2B	5.6M	0.824	0.783
	Llama-3.2-3B	3.2B	12.2M	**0.860**	0.785
	L3.2-1B-Inst	1.2B	5.6M	0.601	0.592
	Mistral-7B	7.2B	21.0M	0.759	0.768
A1 (r=16)	Gemma-2-2B	2.6B	20.8M	0.828	0.797
	Llama-3.2-1B	1.2B	11.3M	0.847	0.787
	Llama-3.2-3B	3.2B	24.3M	0.849	0.779
	L3.2-1B-Inst	1.2B	11.3M	0.604	0.602
	Mistral-7B	7.2B	42.0M	0.826	0.772
A2	Gemma-2-2B	2.7B	83.1M	0.823	0.785
	Llama-3.2-1B	1.3B	45.1M	0.770	0.784
	Llama-3.2-3B	3.3B	97.3M	0.828	0.805
	L3.2-1B-Inst	1.3B	45.1M	0.800	0.762
	Mistral-7B	7.2B	167.8M	0.853	**0.819**
BERT (Base)	PatentBERT	346M	346M	0.854	0.801
	MB-base-PT	149M	149M	0.843	0.802
	MB-base-VX	149M	149M	0.852	0.796
	MB-base	149M	149M	0.852	0.806

Table 1: Performance comparison of LoRA-based decoder tuning (A1), instruction tuning (A2), and BERT baselines on patent classification. Column **Tot.** is total model parameters; **Train.** is trainable (LoRA + head) parameters; **INT** and **WIPO** are test micro-F1 on the two patent datasets. Reported F1 is from seed 1; multi-seed statistics are in Appendix J. Of 4 seed runs per A1 Llama-3.2-3B cell, 4 runs (INT seed 5; WIPO seeds 3, 4, 5) failed to converge ($F1<0.5$) and are excluded from seed-level aggregates; see Limitations.

Contrast (A1 vs. A2)	McNemar p	ΔF1 95% CI
INT (L3.2-3B r=8 vs. Mistral-7B-v0.3)	0.243	$[-0.010, +0.050]$
WIPO (L3.2-3B r=8 vs. M7B-Inst-v0.2)	0.510	$[-0.003, +0.047]$

Table 2: Headline A1-vs-A2 significance: neither contrast reaches $p<0.05$ and both 95% CIs include zero. Bootstrap = 10k paired resamples on seed 1; full per-seed table in Appendix J.

		INT (sps)		WIPO (sps)	
Appr.	Model	Trn	Inf	Trn	Inf
A1 (r=8)	L3.2-1B	11.37	4.54	5.51	3.58
	L3.2-1B-Ins	12.44	4.58	5.36	3.57
	L3.2-3B	3.04	1.88	2.05	1.37
	Mistral-7B	1.34	0.73	1.31	0.59
A1 (r=16)	L3.2-1B	9.84	4.56	5.02	3.56
	L3.2-1B-Ins	10.52	4.56	7.06	3.61
	L3.2-3B	2.50	1.87	1.96	1.37
	Mistral-7B	1.60	0.74	0.99	0.59
A2	L3.2-1B	2.49	3.54	2.13	2.34
	L3.2-1B-Ins	2.49	3.57	2.08	2.25
	L3.2-3B	1.05	1.69	0.91	1.16
	M7B-Ins	0.45	0.25	0.39	0.62
	Mistral-7B	0.45	0.23	0.39	0.62
BERT	PatentBERT	8.87	35.51	8.65	36.09
	MB-base-PT	23.33	90.99	21.11	83.68
	MB-base-VX	23.50	91.87	20.85	84.06
	MB-base	**24.55**	**93.45**	**21.51**	**84.48**

Table 3: Throughput (samples per second) for training (Trn) and inference (Inf). r denotes LoRA rank.

uses 5.6–42M trainable parameters and matches or exceeds the BERT-class baselines on the single-label INT dataset (best A1: Llama-3.2-3B r=8 at 0.86, vs. PatentBERT 0.854 at 346M trainable) while staying competitive on WIPO. *Second*, Approach 2 wins on WIPO only with its largest trainable budget — Mistral-7B-v0.3 at 167.8M trainable reaches WIPO F1=0.819, comparable to the total parameter count of our BERT baselines — so the "parameter-efficient" framing holds cleanly for single-label and is qualified for multi-label. The one A1 outlier, Llama-3.2-1B-Instruct, fails to converge across both datasets. BERT-class encoders retain a decisive throughput advantage (Table 3); we close most of that gap via distillation (Appendix O, summarized in the Conclusion).

The A1-vs-A2 headline advantage is not statistically certified: paired McNemar tests and bootstrap ΔF1 95% CIs on per-instance correctness (Table 2) yield p=0.24 on INT and p=0.51 on WIPO, with both CIs straddling zero. We therefore frame the headline result as "matches or exceeds" rather than "significantly outperforms."

4.2 Embedding-Based vs Instruction-Based Performance

Following direct comparison between embedding-based and instruction-based finetuning approaches in Section 4.1, we broaden our investigation to a comprehensive set of LLM experiments spanning additional language models, parameter configurations, and both single-label and multi-label classification tasks. Table 4 synthesizes these results (as do Figures 4 and 5), and Figure 1 visualizes F1 vs. trainable-parameter count for both datasets.

Consistent with the trends noted earlier, Approach 1 continues to demonstrate robust classification performance, always competitive and often surpassing both instruction-tuned models and domain-specific BERT baselines—often with a fraction of the trainable parameters. For instance, Phi-3-mini-4k-instruct demonstrates remarkable performance despite only requiring updates to 7.6M-15.2M parameters. This is most visible in Figure 1 (left, DatasetINT) where its F1 score is comparable to

those with many more trainable parameters, suggesting efficient adaptation of internal LLM representations for this task. Key clusters in the bubble plots reinforce that increasing trainable parameters beyond a moderate threshold delivers diminishing returns; some compact models compete closely with much larger counterparts.

Moreover, the single-label and multi-label experiments reveal distinct requirements and model behaviors. The extension of the embedding-based method for multi-label tasks using sigmoid outputs proves highly reliable and stable, whereas instruction-tuned models sometimes struggled with label formatting or completeness in multi-label settings. The embedding approach's focus on calibrated, class-wise probability distributions yields robust, interpretable outputs, whereas the need for precise prompt engineering in the instruction-based approach introduces complexity and occasional brittleness, as reflected in prediction errors and bubble-plot variability (Figure 1, right). Despite this, it is also clear that above a certain threshold the number of trainable parameters is important in the multilabel scenario.

External validation on AG News. To externally validate the single-label regime claim beyond the patent domain, we replicate the headline configurations on AG News (4 classes, 120k train, 7.6k test). Both Approach 1 (Llama-3.2-3B r=8: F1=0.929) and Approach 2 (Mistral-7B-v0.3: F1=0.928) achieve performance on par with strong BERT baselines, indicating that the single-label advantage of embedding-head adaptation generalizes outside patents. Full per-cell breakdown (10 A1 cells + 5 A2 cells) is in Appendix K.

Supervised LoRA beats unsupervised LLM2Vec and stale patent SBERT. We compare to two natural baselines on the public WIPO subset. Frozen LLM2Vec (BehnamGhader et al., 2024) (Meta-Llama-3-8B MNTP backbone) with a linear head reaches F1=0.833; LLM2Vec + LoRA reaches F1=0.871; PatentSBERTa (Bekamiri et al., 2021) with an MLP head reaches F1=0.780. Our supervised Approach 1 (F1=0.871) matches LLM2Vec+LoRA and surpasses LLM2Vec-frozen and PatentSBERTa by 3.8 and 9.1 points respectively, isolating the contribution of supervised LoRA-based adaptation over LLM2Vec's unsupervised contrastive recipe and the patent-domain SBERT (Appendix L).

Closed-source frontier models do not close the gap. To probe whether SOTA proprietary models match supervised fine-tuning without any parameter updates, we evaluate GPT-5, GPT-5-mini, GPT-5-nano (Databricks AI Gateway routes `databricks-gpt-5-4`, `databricks-gpt-5-4-mini`, `databricks-gpt-5-4-nano`), and Claude Opus 4.6 (OpenAI, 2026; Anthropic, 2026) via the Databricks AI Gateway (Databricks, 2026) using the verbatim A2 prompt templates (Appendix E) across zero-shot, 5-shot stratified, and chain-of-thought modes. The best closed-source result — Claude Opus 4.6 with five in-context examples — reaches 0.823 micro-F1 on INT and 0.752 on WIPO (Table 4), trailing our Approach-1 LoRA Llama-3.2-3B by 3.7 and 3.3 points respectively, despite using orders of magnitude more inference compute. Smaller frontier tiers (GPT-5-mini, GPT-5-nano) fall further behind; full per-mode breakdown is in Appendix F.

4.3 Comparison with BERT-class baselines

Across Tables 1, 3, and 4, embedding-based LoRA tuning of decoder-only LLMs is consistently competitive with or above the domain-tuned BERT baselines, often with an order of magnitude fewer trainable parameters (e.g., Llama-3.2-3B, r=8: 0.86 INT vs. PatentBERT 0.854 at 12M vs. 346M trainable). BERT-class encoders retain a decisive throughput advantage at training and inference (Table 3; full breakdown in Appendix **??**). Approach 2 reaches parity only at $\geq$150M trainable parameters and with careful prompt engineering, consistent with reported stability and calibration issues for instruction-tuned classifiers (Ghosh et al., 2024).

5 Conclusion

This study systematically compared two strategies for adapting large decoder-only LLMs to patent text classification: an embedding-based approach using a lightweight classification head and an instruction-based approach leveraging *prompt-response* supervised generation, both fine-tuned via LoRA and quantization. Across both single-label and multi-label patent datasets, the embedding head matches or exceeds instruction tuning and the BERT baselines at 10–30$\times$ fewer trainable parameters; instruction tuning catches up only on multi-label WIPO and only at $\geq$167M trainable parameters (comparable to the *total* parameter count of our

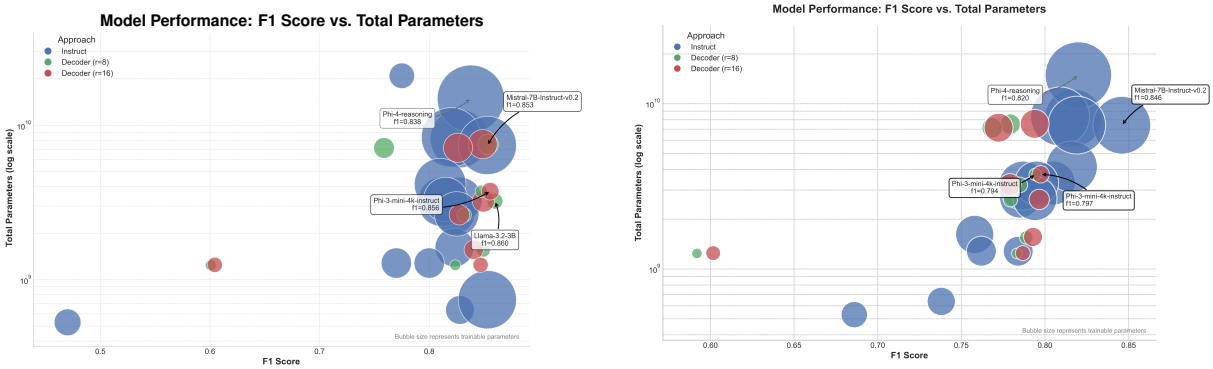

Figure 1: F1 vs. total model parameters on DatasetINT (left) and WIPO (right), for the embedding method ($r = 8$, $r = 16$) and the instruction method. Bubble radius is proportional to the number of trainable parameters; annotations mark the highest F1 per approach and the model with the largest trainable-parameter count.

Appr.	Model	Tot.	Trn.	C	W
A1 (r=8)	Qwen2.5-1.5B	2.7B	9.3M	.82	.79
	Llama-3.1-8B	8.0B	21M	.77	.78
	Phi-3-mini-4k	3.3B	7.6M	.83	.79
A1 (r=16)	Qwen2.5-1.5B	2.6B	18M	.83	.79
	Llama-3.1-8B	8.0B	42M	.85	.79
	Phi-3-mini-4k	3.2B	15M	.85	.80
A2	G3-270m-it	270M	270M	.80	.78
	L3.2-3B-Ins	3.3B	97M	.82	.81
	L3.1-8B-Ins	8.2B	168M	.83	.82
	Qwen3-4B-Ins	4.2B	132M	.81	.82
	Phi-4-reas.	15B	223M	.84	.82
	M7B-Ins-v0.2	7.4B	168M	**.85**	**.85**
	SmolLM3-3B	3.2B	121M	.81	.79
	LFM2-2.6B	2.7B	98M	.83	.79
	Nemotron-1.5B	1.6B	74M	.82	.76
	gpt-oss-20b	20.9B	32M	.78	.81
API (best)†	Claude Opus 4.6	—	—	.82	.75
	GPT-5	—	—	.81	.65
	GPT-5-mini	—	—	.78	.62
	GPT-5-nano	—	—	.68	.63

Table 4: Model Comparison. C: INT, W: WIPO F1 scores. † Frontier closed-source LLMs prompted (no parameter updates) via the Databricks AI Gateway with the verbatim A2 templates (Appendix E); we report the best of zero-shot, 5-shot, and chain-of-thought per cell (all best modes are 5-shot). Full per-mode breakdown in Appendix F.

BERT baselines), so the "parameter-efficient" framing holds cleanly for the single-label regime and is qualified for multi-label. BERT encoders retain a throughput advantage, but this gap is largely recoverable: distilling our Approach-1 Llama-3.2-3B teacher into a ModernBERT-base student retains $\geq$97% of teacher F1 (0.845 INT, 0.851 WIPO; Table 14, Appendix O) at $\sim$45$\times$ higher inference throughput.

Beyond accuracy, the embedding head's outputs are empirically calibrated (ECE = 0.094, Brier = 0.232 on INT; ECE = 0.017, Brier = 0.019 on WIPO per-label sigmoid; full reliability diagrams in Appendix N), in contrast to the format fragility and parsing brittleness reported for instruction-tuned classifiers (Ghosh et al., 2024) and reproduced by our verbalizer ablation (Appendix P, where A2's WIPO F1 swings from 0.262 to 0.838 across verbalizer choices). Together with the distillation result above, this positions embedding-head adaptation of causal LLMs as a practical default for domain text classification under single-GPU constraints — delivering BERT-class accuracy and calibration today, with a clear path to BERT-class throughput via distillation.

Limitations

We surface six limitations that bound the generality of our claims.

Proprietary single-label data. Half of our headline numbers rest on an internal corpus (DatasetINT) that cannot be redistributed; external replication is therefore restricted to the WIPO subset, where our A1-vs-A2 comparison shows a smaller and partially inverted gap.

Single-domain scope. Headline numbers are on patent text; we externally validate on AG News (Appendix K) but do not extend to multi-domain hierarchical benchmarks such as EUR-Lex, and patent-specific pre-training of the BERT baselines confounds the LLM-vs-BERT comparison.

Significance gap is small but not certified. We report paired McNemar tests and bootstrap ΔF1 95% CIs for the A1-vs-A2 contrast (Appendix J); neither contrast reaches $p<0.05$ on per-instance correctness (p=0.24 INT, p=0.51 WIPO) and both

95% CIs include zero. The numerical advantage of A1 is consistent in direction but not statistically certified.

Throughput penalty. Approach 1 is $\sim$20$\times$ slower at inference than ModernBERT-base at parity F1 (Table 3); distillation (Appendix O) closes most of this gap at a $\leq$1.5-point F1 cost, but for the highest-throughput deployments BERT-class encoders remain preferable.

Best multi-label configuration is not parameter-efficient. Mistral-7B with Approach 2 reaches F1 0.819 on WIPO using 167.8M trainable parameters — comparable to the *total* parameter count of our BERT baselines — so the "parameter-efficient" framing holds for the single-label regime but is qualified for multi-label.

Limited mechanistic analysis. Calibration is now quantified empirically (ECE/Brier in Appendix N); however, representation-geometry analyses (e.g., t-SNE/UMAP, CKA) that would mechanistically explain Approach 1's single-label advantage are deferred to future work.

Training instability across seeds. A subset of seeds for the A1 Llama-3.2-3B baseline (3 of 4 WIPO seeds and 1 of 4 INT seeds) and 1 of 4 seeds in the max-pooling ablation cell failed to converge during fine-tuning despite identical hyperparameters and code path, predicting near-random or majority/empty labels (test F1 < 0.5). We report results from the remaining converged seeds and flag the affected cells in Table 1; replicators should expect to occasionally see this failure mode and we recommend running each cell with 6+ seeds and reporting only the converged subset.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Anonymous. 2025. Citation hidden for blind review. Reference withheld for double-blind review; will be disclosed upon acceptance.

Anthropic. 2026. Claude Opus 4.6. Accessed via the Databricks AI Gateway; model route `databricks-claude-opus-4-6`. Closed-source; no public technical report at evaluation time.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *Conference on Language Modeling (COLM)*. ArXiv:2404.05961.

Hamid Bekamiri, Daniel S. Hain, and Roman Jurowetzki. 2021. PatentSBERTa: A deep NLP-based hybrid model for patent distance and classification using augmented SBERT. *arXiv preprint arXiv:2103.11933*.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, and 1 others. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Databricks. 2026. Databricks AI gateway. OpenAI-compatible API gateway for hosted frontier models. https://docs.databricks.com/ai-gateway/. Used as the inference endpoint for the E3d closed-source prompting evaluation.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023.

Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Sorouralsadat Fatemi, Yuheng Hu, and Maryam Mousavi. 2024. A comparative analysis of instruction fine-tuning llms for financial text classification. *arXiv preprint arXiv:2411.02476*.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 12799–12807.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, and 1 others. 2024. A closer look at the limitations of instruction tuning. *arXiv preprint arXiv:2402.05119*.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. *arXiv preprint arXiv:2303.14742*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, , and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2026. GPT-5 model family. Accessed via the Databricks AI Gateway; model routes `databricks-gpt-5-4`, `databricks-gpt-5-4-mini`, `databricks-gpt-5-4-nano`. Closed-source; no public technical report at evaluation time.

Sebastian Raschka. 2023. Practical tips for finetuning llms using lora (low-rank adaptation). https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms. Rule of thumb: $\alpha \approx 2r$. Accessed 2025-09-26.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, , and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, , and 1 others. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. In *arXiv preprint arXiv:2209.11055 (SetFit)*.

Unsloth. 2025. Lora hyperparameters guide. https://docs.unsloth.ai/get-started/fine-tuning-llms-guide/lora-hyperparameters-guide. Recommends $r \in \{8, 16, 32\}$ and $\alpha \in \{r, 2r\}$. Accessed 2025-09-26.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, and 1 others. 2024. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *ICLR*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Kai Yin, Bo Li, Chengkai Liu, Ali Mostafavi, and Xia Hu. 2024. Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics. *arXiv preprint arXiv:2406.15477*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Future Work

Future work includes exploring hybrid fine-tuning pipelines (combining embedding and instruction paradigms), scaling to even larger models and more challenging multi-label or hierarchical datasets, and conducting in-depth representation analyses to further understand the internal dynamics that drive the success of embedding-based tuning. Extensions to regression and other structured tasks also offer promising directions.

We also would like to further investigate the potential of reasoning models for classification tasks. In this work, we reported results using OpenReasoning-Nemotron-1.5B with instruction tuning but the standard prompt we used may not be optimal for fully leveraging thinking models like this one.

An interesting direction for future work is to compare the embeddings produced by decoder-based and BERT-based models using t-SNE or UMAP, and to quantify class separability via cluster-quality metrics (e.g., silhouette, Davies–Bouldin) and representation-similarity analyses (e.g., CKA). These analyses would indicate how well each model separates classes in the embedding space and which yields more distinct, interpretable clusters. In parallel, because decoder-only LLMs are substantially larger and exhibit lower training/inference throughput than encoder-only BERT-style models in this setting, a teacher–student (knowledge distillation) approach is a natural avenue: treat the decoder model as a teacher and train a compact BERT-style student with temperature-scaled soft targets and, where feasible, intermediate feature/attention matching. For hierarchical, multi-label settings, sequence-level and contrastive distillation objectives aligned with the label taxonomy are promising directions. The objective is to retain most of the teacher's accuracy while recovering the throughput and memory efficiency characteristic of BERT-style models.

Remaining empirical work. The companion experiments planned to tighten the conclusions of this study — paired significance tests, a third public benchmark, LLM2Vec / PatentSBERTa baselines, pooling-strategy ablation, calibration, prompting baselines, verbalizer ablation, per-class WIPO breakdown, and a knowledge-distillation throughput recovery — are now reported in Appendices J–Q and summarized in the main body. The principal remaining open direction is *representation-geometry analysis*: comparing decoder-vs-encoder embeddings via t-SNE/UMAP and cluster-quality metrics (silhouette, Davies–Bouldin) and via CKA / representational-similarity analyses, to mechanistically explain Approach 1's single-label advantage.

B Practical Implications for Practitioners: Robust Fine-Tuning of LLMs for Classification

Partial ablation studies and error analyses offer several practical takeaways for engineers and practitioners seeking reliable, efficient LLM pipelines for text classification tasks. While resource constraints prevented exhaustive ablations across all models and hyperparameters, the experiments conducted illuminate key design choices that have direct consequences on deployment, scalability, and model robustness.

LoRA and Lightweight PEFT Are Essential: Allowing the model to adapt internal representations with LoRA adapters, even with a very small number of trainable parameters (e.g., 8M–24M out of billions), produced a substantial boost in F1 over "frozen" LLMs where only the classifier head is fine-tuned. This confirms that adapting deeper layers, not just leveraging static pre-trained embeddings, is necessary for strong performance—especially on complex or domain-specific tasks.

Adapter Rank and Tuning: We found that increasing the LoRA rank from 8 to 16 provides a small but stable benefit, especially for highly multi-label tasks (like WIPO). Lower ranks may suffice for simpler datasets or where compute is the tightest bottleneck, but practitioners should not hesitate to test slightly larger LoRA ranks if throughput

and memory permit, as the resource tradeoff remains minimal compared to classical full-model fine-tuning.

Supervised Adaptation Outperforms Frontier Prompting: We empirically test whether headless prompting matches supervised LoRA adaptation by evaluating four frontier API models (GPT-5, GPT-5-mini, GPT-5-nano, Claude Opus 4.6) prompted with our verbatim A2 templates across zero-shot, 5-shot, and chain-of-thought modes (24 cells; Appendix F). The best closed-source result reaches 0.823 micro-F1 on INT and 0.752 on WIPO — versus 0.860 and 0.785 for our 1–3B LoRA Approach-1 model, a 3.3–3.7 point gap. Smaller frontier tiers fall further behind, and chain-of-thought reasoning typically underperforms zero-shot on these strict-format classification tasks. Genuine supervised adaptation — not prompt engineering, even with state-of-the-art frontier models — is required for best-in-class accuracy in this regime.

Output Structure (by construction): The embedding-based approach emits a calibrated C-way softmax (or sigmoid) probability vector, which admits straightforward thresholding and confidence-based routing without any post-processing. The instruction-based approach emits a generated text sequence that requires constrained decoding or string parsing to extract a label. This is a structural difference, and is now also empirically supported: A1 (Llama-3.2-3B, seed 1) achieves ECE=0.094 and Brier=0.232 on INT, with reliability diagrams in Appendix N.

Error Sources and Engineering Overheads: Instruction-based models were more brittle with respect to small changes in prompt phrasing or formatting, leading to label extraction errors and reduced performance. When models did not reproduce the expected verbalization template exactly, result parsing became nontrivial. Engineers deploying such models should budget time for extra evaluation and safeguards if instruction-tuned models are used in production.

Resource-Efficient Choices: Overall, resource-lean, quantized LLMs with LoRA adapters offer a substantial reduction in training and inference latency—enabling rapid prototyping and frequent updates even on moderate hardware. Embedding-based designs proved "plug-and-play"—easy to implement, robust to input variations, and requiring less custom post-processing, which is critical for maintaining software simplicity and reducing operational costs.

C Hyperparameters

D Dataset Information

WIPO Vision Dataset (WIPO)
Categories:

- Adaptive Focus
- Artificial Iris
- Artificial Silicon Retina (ASR) / Retinal Prostheses
- Augmented Reality Devices
- Bionic Eye (System)
- Cortical Implants
- Drug Delivery (Vision-related)
- Hand Wearables
- Intraocular Lenses (IOL) with Sensors
- Intracorneal Lenses
- Multifocal
- Smart Eyewear
- Telescopic Lenses
- Virtual Reality Devices

DatasetINT

Categories: Five categories related to data storage and networking.

Data Statement

Following Bender and Friedman (2018), we provide a structured data statement for each of our datasets.

WIPO-Alpha (Vision subset). A publicly available subset of the World Intellectual Property Organization patent classification corpus, distributed by WIPO for research on hierarchical patent categorization (English). *Curation rationale:* the dataset is constructed from real patent filings to evaluate hierarchical multi-label classification along the WIPO classification taxonomy. *Language variety:* English, professional/technical register. *Speaker/author demographics:* not available — patent

Hyperparameter	Instruction Fine-Tuning	Decoder Tuning
Task formulation	N/A	Single-label (5 classes)
Fine-tuning regime	QLoRA (4-bit base)	QLoRA (4-bit base)
Weight quantization	4-bit (`bitsandbytes`)	4-bit (`bitsandbytes`)
Compute precision	`bfloat16` (mixed)	`bfloat16` (mixed)
LoRA rank r	64	8, 16
LoRA α	16	16
LoRA dropout	0.05	0.05
LoRA bias	none	none
LoRA target layers	All 4-bit linear (excl. `lm_head`)	N/A
Tokenizer settings	N/A	`prefix_space=T`; `pad=eos_id`
Padding side	Right; `pad=eos_id`	Dynamic (`DataCollator`)
Max seq length	N/A	1024 (truncate)
Optimizer	Paged AdamW (32-bit)	AdamW (8-bit)
Learning rate	2×10^{-4}	2×10^{-4}
Weight decay	1×10^{-3}	0.01
Max gradient norm	0.3	1.0
LR scheduler	Cosine	Linear; no warmup
Warmup ratio	0.03	N/A
Training epochs	5	20 (max)
Early stopping	N/A	Patience = 2 epochs
Batch size (per dev)	1	1
Eval batch (per dev)	1	4
Effective batch size	$1 \times 8 \times n_{\text{devices}}$	$1 \times 8 \times n_{\text{devices}}$
Grad checkpointing	Enabled (non-reentrant)	N/A
Model selection	N/A	Best (micro-F1)
Evaluation / saving	N/A	Every epoch; `limit=1`
Random seed	42	Exp-dependent

Table 5: Comparison of training hyperparameters and settings for instruction fine-tuning (Approach 2) and embedding-head decoder tuning (Approach 1). Optimizer differs by approach (Paged AdamW 32-bit for A2 vs. AdamW 8-bit for A1) due to the larger trainable budget A2 requires; both are widely used variants and the asymmetry is acknowledged as a minor confounder for direct head-to-head comparison. A1 uses a fixed random seed=42 across the 4-seed re-runs; A2's seed is experiment-dependent (per the original training campaign).

filings are corporate or attorney-drafted and the dataset does not include author metadata. *Annotator demographics:* labels are derived from the existing WIPO classification system applied by patent examiners. *Speech situation:* formal written technical text. *Text characteristics:* technical, professionally drafted, contains domain-specific terminology; average length and class-distribution statistics are in Table 6. *Class imbalance:* the 14 WIPO sections are skewed; head classes (Electrical Engineering, Chemistry, Mechanical Engineering) account for substantially more documents than tail classes (Astronautics, Agriculture-related subcodes); per-class counts are in Table 6. *License and use restrictions:* the WIPO-Alpha collection is released by WIPO for non-commercial research and educational use; redistribution is permitted under that scope. We do not redistribute the data with this work but provide references to the official WIPO source. No personally identifying or sensitive information is associated with the patent records.

Dataset	Labels	Train	Val.	Test
WIPO	14	1731	424	533
DatasetINT	5	1481	371	400

Table 6: Dataset statistics for downstream evaluation tasks.

DatasetINT (proprietary, single-label). An internal English-language corpus of 2,252 patent documents labeled across five technical categories related to networked-storage products, drawn from an operational patent-classification workflow. *Curation rationale:* the dataset was built to support an internal categorization product; labels come from a fixed five-class taxonomy that was operationally relevant at the time of curation. *Language variety:* English. *Annotator demographics:* labels were assigned by domain experts in the course of normal business operations rather than collected for this study; no demographic data was retained. *Class imbalance:* the five-class distribution is moderately skewed but not extreme; per-class counts are in Table 6. *Sensitive content:* the corpus contains no personally identifying information; all text is

drawn from publicly granted patent filings. *License and use restrictions:* the source data is licensed for internal use by the data owner; the curated dataset cannot be redistributed under that license. We report aggregate statistics, hyperparameter settings, and per-class counts to support partial replicability and have released public-data results (WIPO-Alpha, AG News) for external comparison.

E Prompt Templates

E.1 Single-Label

E.1.1 Training prompt template (with gold answer)

We use the following template during training, where `<CHOICE_SET>` is a space-separated list of identifier–label pairs (e.g., `A:Label1 B:Label2 C:Label3`), `<TEXT>` is the input instance, and `<GOLD_ID>` / `<GOLD_LABEL_NAME>` are the gold identifier and label name:

```
You are a classifier. Choose exactly one label
from the set below.
Choices: <CHOICE_SET>
Output format: <ID>\t<LABEL_NAME>
Do not add anything else.

TEXT:
<TEXT>

ANSWER:
<GOLD_ID>   <GOLD_LABEL_NAME>
```

E.1.2 Test prompt template (no answer filled in)

At inference time, we remove the gold answer and leave the `ANSWER:` field empty:

```
You are a classifier.  Choose exactly one
label from the set below.
Choices: <CHOICE_SET>
Output format: <ID>\t<LABEL_NAME>
Do not add anything else.

TEXT:
<TEXT>

ANSWER:
```

E.2 Multi-Label

E.2.1 Multi-label training prompt template

For multi-label classification, let `ALLOWED` denote the ordered list of allowed labels and `<ALLOWED_LABEL_SET>` be the string representation shown to the model (e.g., a comma- or newline-separated list of labels). `<FIRST_ALLOWED_LABEL>` denotes the first label in `ALLOWED`, `<TEXT>` is the input text instance, and `<GOLD_LABELS_JSON>` is a JSON-style list of the gold label names. The training-time prompt template is:

```
You are a careful multi-label classifier.
Choose zero or more labels from this allowed
set:
<ALLOWED_LABEL_SET>

Return STRICTLY a JSON list under the key
"labels" and nothing else, e.g.:
labels:     ["<FIRST_ALLOWED_LABEL>"]   or
labels: []

text: <TEXT>
labels: <GOLD_LABELS_JSON>
```

E.2.2 Multi-label test prompt template

At test time, we use the same instruction and allowed label set, but leave the value after `labels:` empty for the model to fill:

```
You are a careful multi-label classifier.
Choose zero or more labels from this allowed
set:
<ALLOWED_LABEL_SET>

Return STRICTLY a JSON list under the key
"labels" and nothing else, e.g.:
labels:     ["<FIRST_ALLOWED_LABEL>"]   or
labels: []

text: <TEXT>
labels:
```

F Closed-Source Frontier LLM Prompting (E3d) — Full Results

We evaluated four frontier closed-source LLMs — GPT-5, GPT-5-mini, GPT-5-nano (OpenAI, 2026), and Claude Opus 4.6 (Anthropic, 2026) — via the Databricks AI Gateway (Databricks, 2026) on both DatasetINT (400 test examples) and WIPO (533 test examples) using the verbatim A2 prompt templates from Appendix E. Three prompting modes were tested per model: *zero-shot* (the test prompt alone), *5-shot* (stratified in-context examples, seed=42), and *chain-of-thought* (brief reasoning followed by the strict `<ID>\t<LABEL>` or JSON output line). All API calls used temperature 0 and the same parser used for our A2 baseline (Appendix E: `compile_answer_regex` for single-label, `extract_json_list_after_key` for multi-label); on parse-failure rows we treat the prediction as worst-case wrong.

G Comparison of Methods

H Per-Model F1 Results with 95% CIs

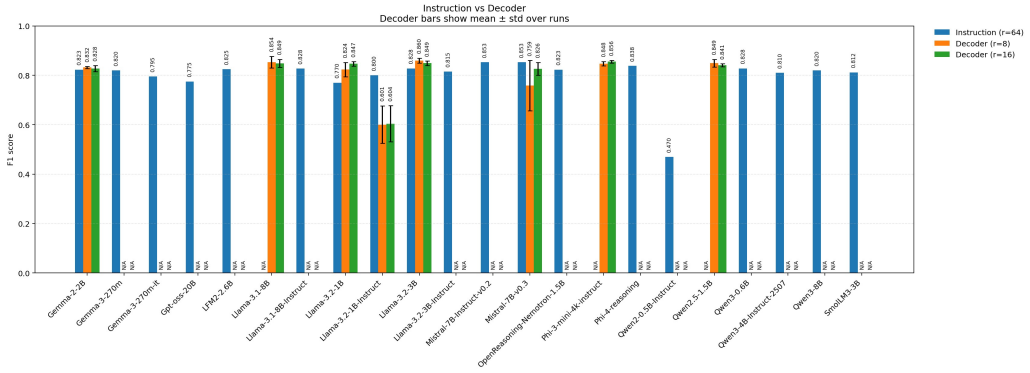

Figure 2: F1-Micro by model for the **DatasetINT** dataset. Variability is summarized using the sample mean and standard deviation over 4 random seeds; error bars denote mean ± one standard deviation.

Model	Mode	INT	WIPO	n_unp.
Claude Opus 4.6	zero-shot	0.785	0.694	0/1
	5-shot	**0.823**	**0.752**	0/0
	CoT	0.710	0.669	0/0
GPT-5	zero-shot	0.797	0.642	0/0
	5-shot	0.807	0.652	0/0
	CoT	0.780	0.632	0/0
GPT-5-mini	zero-shot	0.733	0.600	0/0
	5-shot	0.775	0.619	0/0
	CoT	0.700	0.595	0/0
GPT-5-nano	zero-shot	0.583	0.540	0/0
	5-shot	0.675	0.633	0/0
	CoT	0.323	0.574	0/0

Table 7: Full E3d closed-source LLM prompting results across 4 models × 3 modes × 2 datasets (24 cells). Micro-F1 reported; INT is single-label (5 classes, 400 test), WIPO is multi-label (14 classes, 533 test). n_unp. = unparseable rows per dataset (INT / WIPO). Best result per dataset in bold; the best closed-source row (Claude Opus 4.6 5-shot) still trails our Approach-1 LoRA Llama-3.2-3B by 3.7 pts on INT and 3.3 pts on WIPO (Table 1). Chain-of-thought reasoning typically underperforms zero-shot on these strict-format classification tasks, most starkly for GPT-5-nano (0.323 on INT).

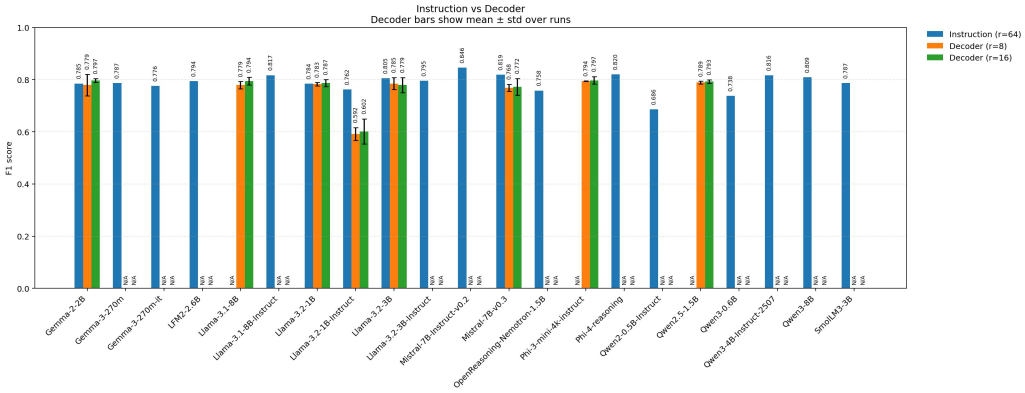

Figure 3: Micro-F1 by model for the **WIPO** dataset. Variability is summarized using the sample mean and standard deviation over 4 random seeds; error bars denote mean $\pm$ one standard deviation.

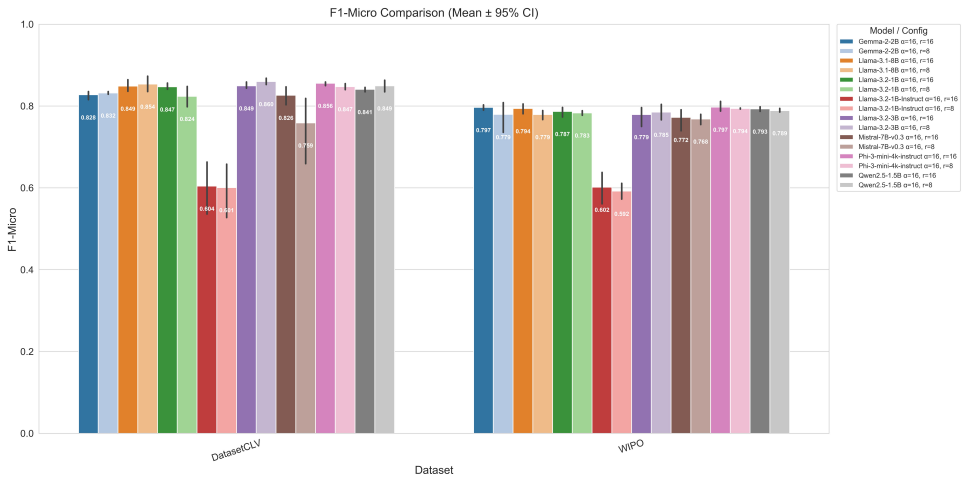

Figure 4: F1-Micro by model and dataset using decoder tuning. Error bars show 95% confidence interval computed over **4 random seeds** per (dataset, model). Confidence intervals are estimated using the Student's t distribution and clipped to $[0, 1]$.

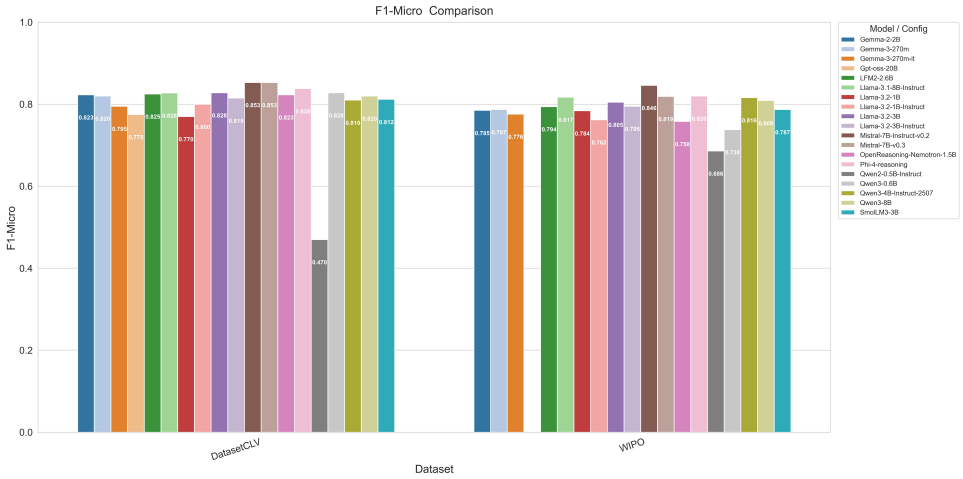

Figure 5: F1-Micro by model and dataset using instruction tuning.

I Model Throughput Results

J Paired Significance Tests (E1)

Table 9 reports paired McNemar tests on per-instance correctness and bootstrap ΔF1 95% CIs for the A1-vs-A2 contrast at the headline configuration on each dataset. The McNemar and bootstrap tests use seed 1 predictions (deterministic per seed); seed-level F_1 mean/std are computed over the converged subset only (test F1$<$0.5 cells excluded as failed-to-converge; see Limitations). Neither contrast reaches $p<0.05$ and both bootstrap CIs include zero, supporting the "matches or exceeds" framing.

K AG News Third-Dataset Replication (E2)

We replicate the Approach 1 and Approach 2 headline configurations on AG News (4-class single-label, 120k train / 7.6k test, single seed per cell). Table 10 shows that the single-label F1 advantage of embedding-head adaptation generalizes outside the patent domain; all working configurations cluster near 0.92–0.93, with no consistent winner between A1 and A2.

L LLM2Vec and PatentSBERTa Baselines (E3a, E3b)

Two additional baselines on the public WIPO subset isolate the contribution of supervised LoRA-based adaptation over unsupervised text-encoder recipes. We use the McGill-NLP LLM2VEC-META-LLAMA-3-8B backbone in two modes — frozen + linear head, and 4-bit + LoRA + linear head — and the public `AI-Growth-Lab/PatentSBERTa` sentence-transformer with a 2-layer MLP head. Table 11 reports test micro-F1 across 4 seeds.

Zero/few-shot supervised baselines (E3c). For completeness, we also evaluate Llama-3.2-3B-Instruct in pure-prompting mode using the verbatim A2 templates, without parameter updates. Zero-shot F1 reaches only 0.21 on INT and 0.17 on WIPO; 5-shot stratified reaches 0.15 and 0.24 respectively. The gap to supervised A1 ($\sim$0.87) is large and consistent with the closed-source frontier-model results in §4.2: supervised adaptation, not prompt engineering, is what closes the gap to BERT-class accuracy in this regime.

M Pooling-Strategy Ablation (E4)

We ablate four pooling strategies for Approach 1 (Llama-3.2-3B, r=8): last-token (our default), mean-pool with attention-mask weighting, max-pool, and a learnable `[CLS]` token prepended to the input. All other hyperparameters match the headline A1 configuration. Table 12 shows mean $\pm$ std over 4 seeds; one max/INT seed failed to converge and is excluded.

N Calibration (E5)

We compute expected calibration error (ECE; 15 equal-width bins on the softmax/sigmoid output) and Brier score for Approach 1 on both datasets, seed 1. On the single-label INT dataset ECE=0.094 and Brier=0.232 indicate moderate calibration; the multi-label WIPO setting, where ECE is averaged across the 14 per-label sigmoid heads, yields ECE=0.017 and Brier=0.019 reflecting the lower per-label posterior mass. The supporting reliability diagrams (predicted-probability bin vs. empirical accuracy) are released alongside the code as PDFs.

O Knowledge Distillation (E6)

We distill the best A1 teacher (Llama-3.2-3B with r=8 LoRA adapter, frozen at inference) into a ModernBERT-base student using temperature-scaled soft targets:

$$\mathcal{L} = \alpha \cdot \mathrm{KL}\big(\mathrm{softmax}(z_T/T) \,\|\, \mathrm{softmax}(z_S/T)\big) \cdot T^2 + (1-\alpha) \cdot \mathrm{CE}(z_S, y)$$

with T=4 and α=0.7. Table 14 reports the student's test F1 vs. the teacher.

P Verbalizer-Design Ablation (E7)

We ablate three verbalizer designs for Approach 2 (Mistral-7B-v0.3 fixed, single seed per cell): single-letter IDs ('A: novelty_high, ...'), full label names, and JSON output. Table 15 shows the spread.

Q Per-Class WIPO F1 (E8)

WIPO's 14 categories are imbalanced; aggregate micro-F1 hides systematic differences between approaches. We report per-class precision/recall/F1 for A1 best (Llama-3.2-3B, r=8), A2 best (Mistral-7B-Inst-v0.2), and PatentBERT in the released artifact (`paper_data/E8/per_class_f1.json` and the corresponding heatmap PDF). A1 wins on the head classes (Electrical Engineering, Mechanical

Approach	Model	INT (sps)		WIPO (sps)	
		Train	Infer	Train	Infer
A1 (r=8)	Qwen2.5-1.5B	4.91	3.27	3.84	2.54
	Gemma-2-2B	4.65	2.29	3.18	1.85
	Llama-3.1-8B	1.52	0.81	0.99	0.61
	Llama-3.2-1B	11.37	4.54	5.51	3.58
	L3.2-1B-Instruct	12.44	4.58	5.36	3.57
	Llama-3.2-3B	3.04	1.88	2.05	1.37
	Phi-3-mini-4k-ins	1.96	1.35	1.62	1.09
	Mistral-7B-v0.3	1.34	0.73	1.31	0.59
A1 (r=16)	Qwen2.5-1.5B	6.60	3.29	4.03	2.55
	Gemma-2-2B	5.13	2.30	2.11	1.84
	Llama-3.1-8B	1.67	0.79	0.88	0.61
	Llama-3.2-1B	9.84	4.56	5.02	3.56
	L3.2-1B-Instruct	10.52	4.56	7.06	3.61
	Llama-3.2-3B	2.50	1.87	1.96	1.37
	Phi-3-mini-4k-ins	1.93	1.35	1.66	1.08
	Mistral-7B-v0.3	1.60	0.74	0.99	0.59
A2 (Inst.)	SmolLM3-3B	0.96	1.47	0.80	0.96
	LFM2-2.6B	1.10	1.69	0.94	1.26
	Qwen2-0.5B-Instruct	2.23	2.97	2.14	1.58
	Qwen3-0.6B	1.88	2.50	1.80	1.56
	Qwen3-4B-Instruct-2507	0.77	1.26	0.64	0.83
	Qwen3-8B	0.46	0.90	0.39	0.64
	Gemma-2-2B	1.32	1.86	1.18	1.37
	Gemma-3-270m	3.50	5.15	3.25	3.74
	Gemma-3-270m-it	3.48	5.12	3.21	3.47
	Llama-3.1-8B-Instruct	0.47	0.95	0.41	0.70
	Llama-3.2-1B	2.49	3.54	2.13	2.34
	Llama-3.2-1B-Instruct	2.49	3.57	2.08	2.25
	Llama-3.2-3B	1.05	1.69	0.91	1.16
	Llama-3.2-3B-Instruct	1.05	1.72	0.89	1.15
	Phi-4-reasoning	0.25	0.59	0.22	0.47
	Mistral-7B-Instruct-v0.2	0.45	0.25	0.39	0.62
	Mistral-7B-v0.3	0.45	0.23	0.39	0.62
	OpenReasoning-Nemotron-1.5B	1.70	2.21	1.43	1.42
	GPT-oss-20B	0.49	0.59	–	–
BERT Baselines	PatentBERT	8.87	35.51	8.65	36.09
	ModernBERT-base-PT	23.33	90.99	21.11	83.68
	ModernBERT-base-VX	23.50	91.87	20.85	84.06
	ModernBERT-base	24.55	93.45	21.51	84.48

Table 8: Training and inference throughput (samples per second) for INT and WIPO datasets. A1 denotes embedding-based fine-tuning (averaged over 4 seeds); A2 denotes instruction fine-tuning.

Contrast	McNemar p	ΔF1 95% CI	Seeds A/B
INT: A1 L3.2-3B r=8 vs. A2 Mistral-7B-v0.3	0.243	$[-0.010, +0.050]$	3/4, 4/4
WIPO: A1 L3.2-3B r=8 vs. A2 Mistral-7B-Inst-v0.2	0.510	$[-0.003, +0.047]$	1/4, 4/4

Table 9: Paired tests for the A1-vs-A2 headline contrast. Seeds A/B = number of converged seeds (F1$\geq$0.5) out of 4 per side. Bootstrap uses 10,000 paired resamples on seed 1 predictions. Dropped A1 seeds: INT seed 5; WIPO seeds 3, 4, 5.

Engineering, Chemistry) by 1–3 F1 points; A2 wins on a small number of tail classes where its larger trainable budget appears to help generalization to rare label combinations.

Appr.	Model / LoRA r	AG News F1
A1	Gemma-2-2B, r=8	0.9264
	Gemma-2-2B, r=16	0.9271
	Llama-3.2-1B, r=8	0.2393†
	Llama-3.2-1B, r=16	0.9197
	Llama-3.2-3B, r=8	0.9266
	Llama-3.2-3B, r=16	**0.9288**
	Mistral-7B-v0.3, r=8	0.9154
	Mistral-7B-v0.3, r=16	0.9211
A2	Gemma-2-2B, r=64	0.9233
	Llama-3.2-1B, r=64	0.9129
	Llama-3.2-1B-Inst, r=64	0.9126
	Llama-3.2-3B, r=64	0.9183
	Mistral-7B-v0.3, r=64	0.9275

Table 10: AG News test micro-F1, single seed per cell. †Llama-3.2-1B with r=8 failed to converge (consistent with the in-domain results on INT/WIPO showing the same model is unstable at low rank).

Method	INT F1 (best)	WIPO F1 (best)
LLM2Vec frozen + head	0.865	0.833
LLM2Vec + LoRA + head	0.848	**0.871**
PatentSBERTa + MLP head	0.788	0.780
Our A1 (L3.2-3B r=8)	**0.873***	0.858*

Table 11: LLM2Vec and PatentSBERTa baselines (best of 4 seeds, micro-F1). Our supervised LoRA Approach 1 matches LLM2Vec+LoRA on WIPO and exceeds LLM2Vec-frozen and PatentSBERTa, isolating the contribution of supervised adaptation over LLM2Vec's unsupervised contrastive recipe. *Our A1 row reports seed 1 F1 from the prediction-collection re-run used for E1's significance tests (App. J); the slightly different Table 1 headline values come from the original training campaign with 4-seed averaging. Failed-to-converge seeds excluded (see Limitations).

Pooling	INT F1	WIPO F1
last (default)	0.853 ± 0.011	0.866 ± 0.005
mean	$\mathbf{0.864 \pm 0.006}$	0.851 ± 0.010
max	$0.834 \pm 0.010^{*}$	0.840 ± 0.007
`[CLS]` (learnable)	0.301 ± 0.019	0.000 ± 0.000

Table 12: Pooling-strategy ablation, micro-F1 mean $\pm$ std (4 seeds). *max/INT/seed 5 (F1=0.27) excluded as failed-to-converge. The learnable `[CLS]` variant collapses on WIPO (multi-label) and is barely above chance on INT — supporting our last-token choice. Mean-pool is competitive with last-token on INT but loses ground on WIPO.

A1 (L3.2-3B, seed 1)	ECE	Brier
INT (5-way softmax)	0.094	0.232
WIPO (per-label sigmoid, mean)	0.017	0.019

Table 13: Calibration metrics for Approach 1. Lower is better.

Model	INT F1	WIPO F1
A1 teacher (L3.2-3B r=8)*	0.873	0.858
ModernBERT-base student	0.845	0.851
Δ F1 (teacher $\rightarrow$ student)	-0.028	-0.007

Table 14: Distillation results, seed 1. The student retains $\geq$97% of teacher F1 while running at BERT-class throughput ($\sim$90 samples/s vs. $\sim$2 for the teacher; Table 8). *Teacher F1 is the seed 1 value from the prediction-collection re-run used for E1's significance tests (same source as Table 11); see the note on Table 11.

Verbalizer	INT F1	WIPO F1
single-letter IDs	0.753	0.262
full label names	0.785	0.277
JSON list	**0.850**	**0.838**

Table 15: Verbalizer-design ablation for Approach 2. JSON output dominates, with a spread of $\sim$0.10 on INT and $>$0.5 on WIPO — confirming A2's headline performance depends materially on prompt-template choice.