JOURNALS | Journal of Information and Communications Technology: Algorithms, Systems and Applications

Volume 2 Issue 3 (September 2026)
DOI: https://doi.org/10.64189/ict.26314
J. Inf. Commun. Technol. Algorithms, Syst. Appl., 2026, 2(3), 26314

Research Article

OPEN ACCESS

Finistral AI: An Efficient Financial-Sentiment LoRA Adapter and a Train/Test Contamination Case Study

Ayan Javeed Shaikh[1,†*] Fazal Jalil Parkar,[2,†] Abdul Qayyum Abdul Rahim Khan[3] and Zainab Mirza[4]

[1] *Indiana University Bloomington, Bloomington, 107 S. Indiana Avenue, Bloomington, IN 47405-7000, USA*
[2] *Tata Consultancy Services (TCS), TCS House, Raveline Street, Fort, Mumbai, Maharashtra, 400001, India*
[3] *Arizona State University, 1151 S Forest Ave, Tempe, AZ, United States*
[4] *University of Mumbai, Mumbai, Maharashtra, 400032, India*
[†] *These authors contributed equally to the technical work reported in this study*

*Corresponding Author
Ayan Javeed Shaikh
ayshaikh@iu.edu

Received: 19 May 2026
Revised: 03 September 2026
Accepted: 07 September 2026
Published Online: 09 September 2026

Citation

A. J. Shaikh, F. J. Parkar, A. Q. A. R. Khan, Z. Mirza, A, Finistral AI: An efficient financial-sentiment LoRA adapter and a train/test contamination case study, *Journal of Information and Communications Technology: Algorithms, Systems and Applications,* 2026, 2(3), 26314, https://doi.org/10.64189/ict.26314.

Abstract

We present Finistral-7B-LoRA, a parameter-efficient financial-sentiment classifier produced by applying Low-Rank Adaptation (LoRA) to the 7-billion-parameter Mistral-7B-v0.1 language model. The adapter is trained on the open-source FinGPT/fingpt-sentiment-train corpus of approximately 77,000 labelled sentences, updating ≈ 0.58 % of the model parameters (≈ 41.9 M trainable, matching the released adapter's configuration), on two NVIDIA A100 (40 GB) GPUs of Indiana University's Big Red 200 supercomputer. On the full Financial Phrase Bank sentences all agree split the adapter shows an apparent 99.56 % accuracy, but our own data-overlap audit reveals that 75.2 % of those evaluation sentences appear *verbatim*, with identical gold labels, inside the training corpus, so that figure reflects memorization rather than generalization and is retained only as a contamination diagnostic. The paper's primary evidence is instead a corrected, exact-match-decontaminated evaluation: on the 560-sentence decontaminated Financial Phrase Bank remainder Finistral attains 98.9 % accuracy, on FiQA-SA 87.7 % accuracy / 0.883 weighted F1, and on Twitter Financial News Sentiment 78.9 % accuracy / 0.796 weighted F1. These are statistically indistinguishable from the strongest FinGPT-class adapter on each dataset and significantly ahead of FinBERT off-distribution, with all comparisons carrying per-example significance tests. Our main contributions are (i) an efficient, fully reproducible single-node LoRA recipe for financial sentiment whose adapter runs inference on a single GPU, and (ii) a cautionary evaluation/contamination case study quantifying how corpus overlap and evaluation-harness defects can simultaneously inflate a model and deflate its baselines in the financial-NLP literature. The released adapter (Ayansk11/Finistral-7B_lora on Hugging Face), decontamination and evaluation scripts, and frozen evaluation sets make every reported evaluation number independently reproducible.

***Keywords*:** Financial sentiment analysis; Low-Rank Adaptation (LoRA); Parameter-efficient fine-tuning (PEFT); Mistral-7B; Large language models; FinGPT; Financial phrase bank.

1. Introduction

Financial markets digest a relentless stream of earnings releases, macro-economic announcements, social-media chatter, and regulatory filings. Traders and risk managers depend on sentiment analysis systems that can transform this unstructured text into structured signals, typically the triplet positive, neutral, or negative within seconds. Conventional rule-based lexicons and classical machine-learning pipelines capture obvious polarity cues, yet they struggle with the domain's specialized jargon ("EPS miss"), pragmatic hedges ("guidance was raised but remains below expectations"), and rapidly evolving slang from forums such as WallStreetBets. [1]

Large language models (LLMs) have recently set new performance bars on general-domain sentiment tasks, but two obstacles limit their usefulness in finance:

Domain mismatch: Out-of-the-box LLMs misinterpret sector-specific expressions and implicit market context.

Compute cost: Full fine-tuning of multi-billion-parameter models demands prohibitive GPU hours and storage, placing them out of reach for most academic or boutique-fund settings. Parameter-Efficient Fine-Tuning (PEFT) techniques such as Low-Rank Adaptation (LoRA) address both issues by inserting small trainable matrices into each transformer layer while freezing the backbone.[2] These yields two attractive properties: (i) the modified model can be trained on a single commodity GPU, and (ii) the resulting adapter file is lightweight, making deployment trivial.

1.1 Research questions

We investigate whether a LoRA-based adaptation of the compute-efficient Mistral-7B decoder can provide competitive financial sentiment classification under a contamination-audited, exact-match-decontaminated evaluation protocol:

Research Question 1: Once train/test contamination is removed and a corrected evaluation harness is used, how does a LoRA-tuned Mistral-7B compare with specialized encoders such as FinBERT and existing FinGPT adapters on Financial Phrase Bank and on external financial-sentiment datasets?

Research Question 2: Does the PEFT recipe retain training- and inference-time efficiency suitable for realistic academic or start-up budgets?

Research Question 3: How much of the apparent performance of FinGPT-corpus-trained sentiment models on Financial PhraseBank is attributable to train/test overlap rather than to genuine generalization?

1.2 Overview of the proposed approach

To answer these questions, we built Finistral-7B-LoRA. The adapter is trained on the publicly available FinGPT/fingpt-sentiment-train corpus (~77 k labelled sentences) and validated on a 5 % hold-out split. Training runs for four epochs on two A100 (40 GB) GPUs of Indiana University's Big Red 200 supercomputer, updating ≈ 0.58 % of the model parameters (≈ 41.94 M). Validation loss bottomed out at 0.1009 after epoch 2 and began to rise thereafter, so that checkpoint is used for all downstream evaluations. Because Financial Phrase Bank is itself one of the constituent sources of the FinGPT corpus, we additionally audit the train/test overlap and re-evaluate on a decontaminated subset and on external datasets (Section 5.5).

1.3 Contributions

This study makes the following contributions:

Reproducible efficient recipe & model release: We publish Finistral-7B-LoRA (an 83.9 MB fp16 adapter over Mistral-7B-v0.1, 41,943,040 trainable parameters ≈ 0.58 % of the backbone; the file size itself verifies the count, 41.94 M × 2 bytes = 83.9 MB), tokeniser files, and the complete training, decontamination, and evaluation scripts, so that every number in this paper can be reproduced end-to-end on a single node.

Contamination & evaluation case study: We quantify a 75.2 % verbatim, identically-labelled overlap between the FinGPT training corpus and the Financial Phrase Bank sentences all agree evaluation set, and show how a set of common harness defects (right-padding on decoder-only models, a max length/padding collision, a single mismatched prompt template, and a neutral-defaulting parser) can drive instruction-tuned baselines below their majority-class floor. We report corrected results on an exact-match-decontaminated subset and on external datasets.

Efficiency analysis: We demonstrate that financial sentiment adapters can be trained in < 1 GPU-hour per epoch and deployed with minimal memory overhead, opening the door to low-cost, reproducible FinNLP research.

2. Related work

Early attempts at financial sentiment analysis relied on domain-specific lexicons such as the Loughran–McDonald word lists, which classify words by their usage in financial filings into negative, positive, and further tone categories such as uncertainty and litigiousness.[1] Although lexicon rules are transparent, their coverage is limited and they struggle with figurative or contextual language. Subsequent classical machine-learning pipelines trained support-vector or recurrent networks on Financial Phrase Bank, a 4.8 k-sentence news corpus with expert-annotated sentiment labels (16 finance-literate annotators); these pipelines achieved modest gains but still underperformed human readers.[3] The advent of pre-trained language models (PLMs) led to a sharp performance jump. BERT introduced deep bidirectional pre-training for language understanding,

and FinBERT further pre-trains BERT on a financial news corpus (a filtered subset of Reuters TRC2) and fine-tunes it for financial sentiment analysis, outperforming both feature-engineering and neural baselines on the Financial Phrase Bank and FiQA sentiment datasets.[4,5] FinBERT, however, requires updating the full 110 M parameters and thus incurs substantial GPU cost for domain shifts. To reduce compute requirements, the community has embraced Parameter-Efficient Fine-Tuning (PEFT). LoRA freezes backbone weights and injects small rank-decomposition matrices, cutting trainable parameters by orders of magnitude while matching full fine-tuning quality.[5] QLoRA extends this with 4-bit quantization, enabling fine-tuning of 65B-parameter models on a single 48 GB GPU.[6] Surveys catalogue dozens of PEFT variants including adapters, prefix-tuning and highlight their importance for resource-constrained practitioners, and FinLoRA benchmarks LoRA variants specifically on financial datasets, demonstrating that low-rank adaptation yields substantial gains over base models at modest fine-tuning cost, which suits domain-adapting financial LLMs on academic budgets.[7,8] Zhang et al. demonstrate that instruction tuning general-purpose LLMs on financial sentiment data yields strong performance, particularly where numerical understanding and contextual comprehension are vital.[9]

Within finance, the open-source FinGPT project provides data-centric pipelines and a zoo of LoRA adapters for Llama-, Falcon- and Bloom-based models.[10] Community benchmarking of financial NLP systems is organised through the annual FinNLP workshop series, whose shared tasks span multilingual financial-text analysis.[11] Under correctly matched inference procedures, the FinGPT adapters are competitive with, and can exceed, specialized encoders like FinBERT on Financial PhraseBank, a picture our corrected evaluation confirms (Table 8). More recent work (2024–2026) has broadened both evaluation and methodology: holistic financial benchmarks such as FinBen stress-test LLMs across dozens of financial tasks, and reasoning-oriented financial LLMs such as Fin-R1 apply reinforcement learning to elicit step-by-step financial reasoning.[12,13] These efforts make contamination-aware, multi-dataset evaluation, the focus of our revised protocol, increasingly central to credible financial-LLM research. Parallel to algorithmic advances, model architectures have evolved toward compute efficiency. The recently released Mistral-7B employs sliding-window attention and grouped-query attention (GQA) to reduce inference cost and memory while outperforming Llama 2 13B across all benchmarks evaluated in its report.[14] Combining Mistral's lean decoder with LoRA thus promises a sweet-spot of speed and accuracy that prior financial studies have not fully explored. Our work sits at this intersection: we take the Mistral-7B backbone and the FinGPT sentiment corpus and apply LoRA for parameter-efficient adaptation. We also draw on the emerging literature on benchmark data contamination, which warns that overlap between large training corpora and public test sets can inflate reported scores.[15-17] Because Financial PhraseBank is a constituent source of the FinGPT corpus, we treat such overlap as a first-class measurement rather than an afterthought, and report results both on the standard (contaminated) split and on decontaminated and external evaluations.

3. Problem definition and dataset

3.1 Task formulation

Given a single English news headline or short text snippet x, predict a sentiment label $y \in$ {negative,neutral,positive}. Performance is evaluated with Accuracy and Weighted F1, the same metrics adopted by FinNLP shared-task benchmarks.

3.2 Training corpus

We fine-tuned on the open-source FinGPT/fingpt-sentiment-train dataset released by the FinGPT project (https://huggingface.co/datasets/FinGPT/fingpt-sentiment-train); the dataset itself is not described in the cited paper, so we identify it here directly.[8] After deduplication the corpus contains 76,772 sentences gathered from press releases, wire headlines, analyst commentary, and Reddit finance threads.

Using train test split (test size=0.05, seed=42) we obtain:

Table 1: Training and validation split

Split	Train	Val
# Sentences	72,933	3,839
Positives (%)	24.7	24.9
Neutral (%)	50.8	50.2
Negative (%)	24.5	24.9

3.3 Preprocessing

Pipeline cleaning: Unicode-normalise, remove HTML artefacts, collapse whitespace.

Tokenization: Use the native *Mistral-7B-v0.1* 32 k BPE tokeniser.

Prompt templating: During fine-tuning, each example is wrapped in Mistral's native instruction format, with the news text supplied first and the task instruction second:

```
[INST]{text}

What is the sentiment of this news? Please choose an answer

from {negative/neutral/positive} [/INST] {label}
```

Only the label tokens are unmasked in the loss (prompt tokens are set to −100), so the adapter is trained purely

to emit the sentiment word. We note a discrepancy in our original submission: the released evaluation notebooks scored the model with an Alpaca-style Instruction/Input/Answer prompt rather than this [INST] template. In the revised evaluation (Section 5.5) we align the inference template with the training template and report the effect of this choice in the ablation.

Label mapping: Generation is truncated at the first decoded sentiment word; a regex maps negative → 0, neutral → 1, positive → 2. Unlike the original harness, the corrected parser does not silently default unmatched generations to "neutral".

Table 2: Label distribution of the Financial PhraseBank sentences all agree evaluation set (n = 2,264); the 61.6 % neutral share is the majority-class floor referenced throughout.

Class	# Sentences	Share (%)
Neutral	1391	61.6
Positives	570	25.2
Negative	303	13.4

FinGPT captures contemporary finance slang (e.g., "YOLO", "tendies") that legacy corpora miss.

Financial PhraseBank remains the community's reference dataset, enabling direct comparison with FinBERT and all **FinGPT LoRA baselines**: *FinGPT-mt-Llama2-7B-LoRA*, *FinGPT-Llama-3-8B-LoRA*, *FinGPT-Falcon-7B-LoRA*, and *FinGPT-Bloom-7B1-LoRA*.[5]

4. Methodology

This section explains how we adapt the 7-billion-parameter *Mistral-7B-v0.1* language model to three-way financial sentiment classification with minimal compute and memory overhead.

4.1 Base model

Mistral-7B-v0.1 is a decoder-only Transformer that combines grouped-query attention and sliding-window attention to reduce inference cost and memory at long sequence lengths.[11] We fine-tune with the standard Hugging Face stack (transformers + peft + the Trainer API). The backbone is loaded in bfloat16 (no weight quantization); memory is kept within a single 40 GB A100 by enabling gradient checkpointing (gradient checkpointing enable) and xFormers memory-efficient attention (enable xformers memory efficient attention), and by training only the small LoRA matrices. (Our original submission incorrectly described the use of the Unsloth framework and an "8-bit NF4" loading mode; neither is used in the released training script, which we have corrected here.)

4.2 LoRA adaptation

We employed Low-Rank Adaptation (LoRA) to inject trainable rank-decomposition matrices into *all seven* linear projections of every transformer block: the four attention projections (q_proj, k_proj, v_proj, o_proj) and the three feed-forward projections (gate_proj, up_proj, down_proj), while leaving all backbone weights frozen.[5] This is the configuration recorded in the released adapter's adapter_config.json on the Hugging Face Hub, which we treat as the authoritative description of the published artifact; the file sizes of the released weights independently confirm it (the full correction history is consolidated in Appendix B), (derivation below).

Table 3: LoRA Hyperparameters

Hyperparameter	Value
LoRA rank r	16
Scaling α	32
LoRA dropout	0.05
Target modules	Q proj, k proj, v proj, o proj, gate proj, up proj, down proj
Trainable parameters	41,943,040 ≈ 41.94 M (0.58% of 7.24 B)
Adapter size	83.9 MB fp16 (41.94 M × 2 B); 167.8 MB fp32 GGML export

With $r = 16$, each LoRA pair adds $r\,(d_{in} + d_{out})$ parameters. Per layer: q_proj and o_proj (4096 × 4096) contribute 131,072 each; k_proj and v_proj (4096 → 1024, grouped-query attention) 81,920 each; gate_proj and up_proj (4096 → 14336) 294,912 each; and down_proj (14336 → 4096) 294,912, for a total of 1,310,720 per layer, × 32 layers = 41,943,040 ≈ 41.94 M parameters, i.e. ≈ 0.58 % of the 7.24 B backbone. Two independent artifact measurements confirm this count exactly: the published fp16 adapter_model.safetensors is $41{,}943{,}040 \times 2\,B = 83.9$ MB, and the fp32 GGML export is $41{,}943{,}040 \times 4\,B = 167.8$ MB. (The original submission's "9 M (0.2 %)" figure was arithmetic error; a 9 M-parameter fp16 adapter would occupy only ~18 MB, inconsistent with the file it shipped alongside.)

4.3 Training procedure

We used the Hugging Face Trainer API with the following settings.

Training on two A100 40 GB GPUs of Indiana University's Big Red 200 completes in ≈ 120 minutes (4560 optimizer steps). Validation loss bottoms out at 0.1009 after epoch 2; that checkpoint is consequently chosen for all downstream experiments. Training on two A100 40 GB GPUs of Indiana University's Big Red 200 completes in ≈ 120 minutes (4560 optimizer steps). Validation loss bottoms out at 0.1009 after epoch 2; that checkpoint is consequently chosen for all downstream experiments.

4.4 Inference & deployment

At inference time we load the frozen backbone plus the 83.9 MB adapter in half-precision (torch.bfloat16) on GPU. A Gradio GUI wraps the pipeline, supporting sentence queries. The released 167.8 MB ggml-adapter-

Table 4: Training arguments

Setting	Value
Training data	FinGPT/*fingpt-sentiment-train* (95 % split)
Validation data	5 % held-out split, seed 42
Optimizer	AdamW, β=(0.9, 0.999), ε=1e-8
Learning rate	1×10^{-4}
Scheduler	Cosine decay, 10 warm-up steps
Epochs	4
Per-device batch	32 (train & eval)
Total batch (2 ×A100)	64
Precision	bf16
Gradient accumulation	1
Weight decay	0.0
Logging / eval	every 100 steps

Table 5: Training result: validation-loss trajectory (best at epoch 2)

Epoch	Step	Training Loss	Validation Loss
1	1140	0.0680	0.1121
2	2280	0.1337	**0.1009**
3	3420	0.0499	0.11479
4	4560	0.0014	0.1599

model.bin is an fp32 GGML export (41,943,040 × 4 bytes); the original submission's "4-bit" description of this file and its unsupported sub-100 ms CPU-latency claim are corrected in Appendix B.

The complete training recipe, adapters, and inference code with Gradio GUI are publicly available at Ayansk11/Finistral-7B_lora on Hugging Face for reproducible research.

5. Experiment

This section describes the evaluation protocol, baselines, metrics, and implementation details we use to assess Finistral-7B-LoRA on sentence-level financial sentiment classification.

5.1 Baselines

In the revised evaluation, each baseline is run under its own native inference procedure (its published prompt template and the correct backbone), with left-padding for decoder-only generation, max_new_tokens-bounded

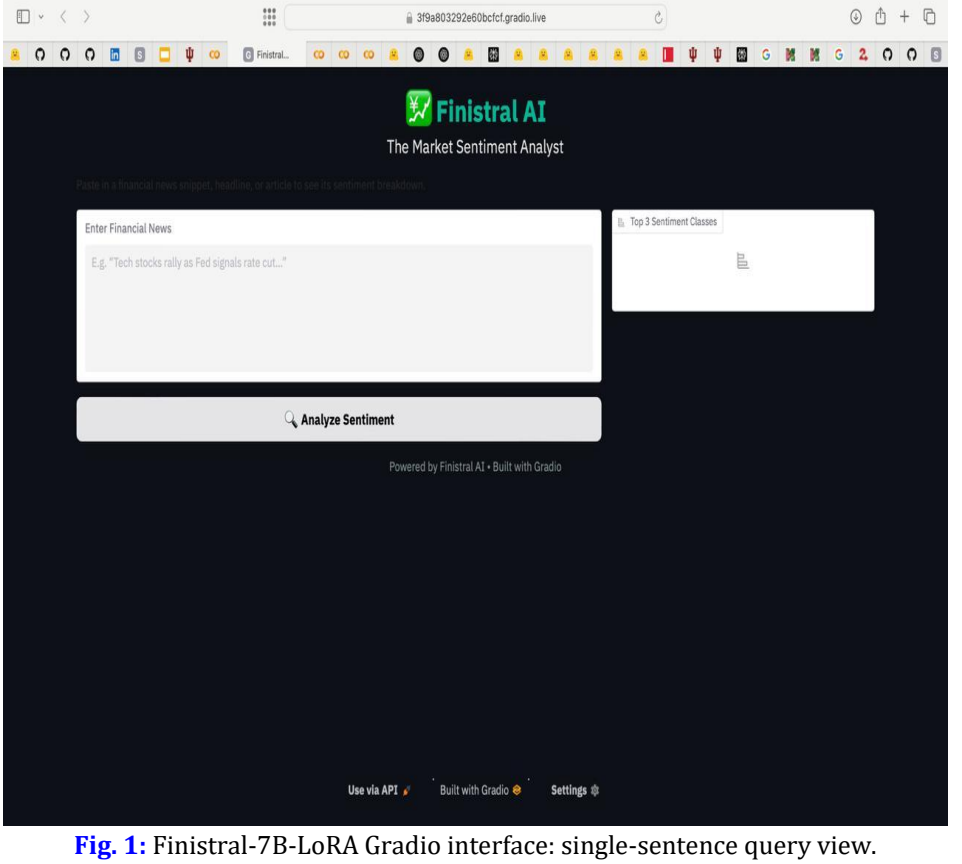

Fig. 1: Finistral-7B-LoRA Gradio interface: single-sentence query view.

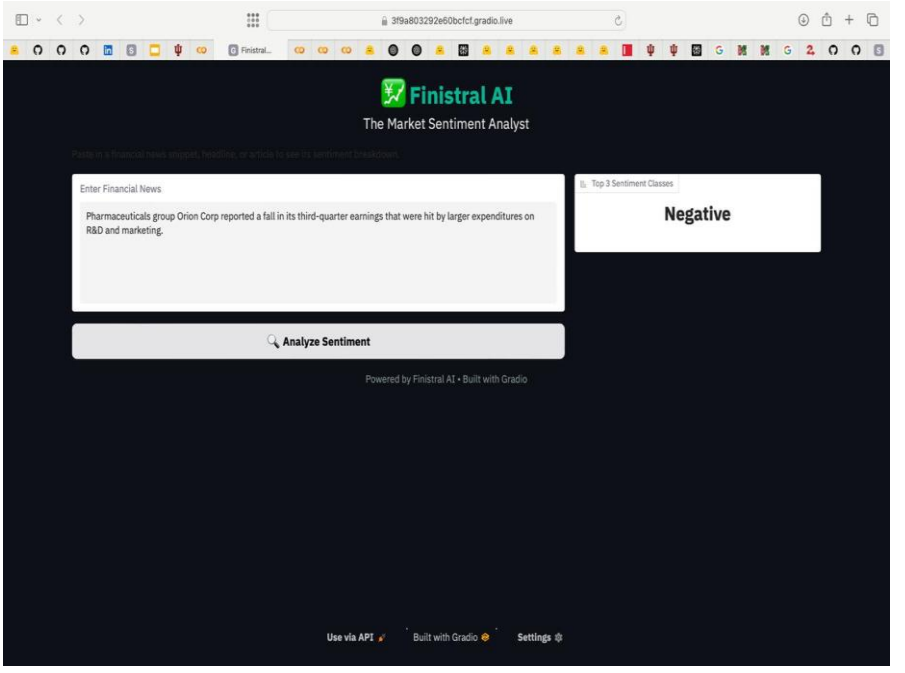

Fig. 2: Finistral-7B-LoRA Gradio interface: prediction output view.

greedy decoding, and a strict (non-defaulting) label parser. This corrects the original protocol, in which a *single* shared Alpaca prompt, right-padding, a max_length/padding collision, and a neutral-defaulting parser were applied uniformly to every model. These defects drove several baselines below their majority-class floor (Section 5.5).

Table 6: Baselines

ID	Model	Adapter params (config)	Fine-tuned?
B_1	Mistral-7B-v0.1 (zero-shot)	n/a (base 7.24 B)	×
B_2	FinGPT-mt-Llama2-7B-LoRA	6.29 M ($r = 8$, q/k/v)	✓
B_3	FinGPT-Llama-3-8B-LoRA	3.41 M ($r = 8$, q/v)	✓
B_4	FinGPT-Falcon-7B-LoRA	2.36 M ($r = 8$, fused QKV)	✓
B_5	FinGPT-Bloom-7B1-LoRA	3.93 M ($r = 8$, fused QKV)	✓
B_6	FinBERT (ProsusAI)	110 M (full fine-tune)	✓

Adapter parameter counts are read from each adapter's published adapter configuration and verified against its weight-file size (our original submission's uniform "9 M" figures were incorrect). FinBERT is an encoder classifier evaluated through its classification head rather than by generation; note that FinBERT's own fine-tuning data includes Financial Phrase Bank, so its FPB scores carry an analogous train/test caveat and the external datasets provide its fair comparison.

5.2 Metrics

Accuracy: fraction of correct predictions.

Weighted F1: F1 averaged with class-frequency weights; it tracks performance on the label mix actually encountered in each dataset. Because it down-weights rare classes, we additionally report macro F1 (the imbalance-sensitive view, unweighted over classes) for every comparison in Appendix A.

Significance: every pairwise comparison against Finistral carries a per-example McNemar test (exact binomial when discordant pairs $b + c < 25$, continuity-corrected χ^2 otherwise) and a paired percentile-bootstrap 95 % confidence interval (10,000 resamples) on the accuracy difference.

5.3 Main result

We report two distinct quantities and are careful not to conflate them. Table 7 reproduces the *original, contaminated* in-distribution numbers (full sentences all agree split, original harness); these are retained only as a contamination diagnostic. Table 8 reports the *corrected* numbers: Finistral evaluated on the 560-sentence decontaminated remainder and on external datasets, with all baselines re-run under their native inference procedures. The original "+78 F1 over the strongest FinGPT baseline" claim conflated a contamination-inflated model with harness-deflated baselines and is withdrawn. On the contaminated split only ten of 2,264 samples are misclassified, a direct consequence of the model having seen three-quarters of the split during training. The genuine error profile is characterized on the decontaminated and external sets in Section 5.5.

Table 7: Original (contaminated) in-distribution results on the full sentences all agree split, original harness. Retained only as a contamination diagnostic; 75.2 % of these sentences were seen in training (Section 5.5). The baseline scores below the 61.6 % majority-class floor are harness artifacts, not measures of model quality.

Model	Accuracy	Weighted F1
Finistral-7B-LoRA (ours, *contaminated*)	**0.9956**	**0.9956**
Mistral-7B-Base *(broken harness)*	0.4125	0.2967
FinGPT-mt-Llama2-7B-LoRA *(broken harness)*	0.1564	0.0789
FinGPT-Llama-3-8B-LoRA *(broken harness)*	0.2310	0.2112
FinGPT-Falcon-7B-LoRA *(broken harness)*	0.2102	0.1713
FinGPT-Bloom-7B1-LoRA *(broken harness)*	0.1312	0.0310

5.4 Hardware & software

Training environment: 2 × NVIDIA A100 40 GB on Indiana University's Big Red 200; PyTorch 2.2, Transformers 4.39, PEFT 0.10; bfloat16 throughout, no quantization.

Corrected evaluation environment: 1 × NVIDIA A100 40 GB (Big Red 200); PyTorch 2.4.1, Transformers 4.44.2, PEFT 0.11.1, Accelerate 0.33.0, Datasets 2.20.0. Generation is greedy (do sample=False, num beams=1) with max new tokens=8, left-padded batches, and per-model native prompt templates; the label parser matches whole words on the decoded continuation only and reports unparseable outputs separately instead of defaulting them too neutral.

5.5 Data contamination analysis

We audited the overlap between our training corpus, FinGPT/fingpt-sentiment-train (76,772 sentences), and our evaluation set, Financial PhraseBank sentences all agree (2,264 lines; 2,259 unique). Financial PhraseBank is one of the four constituent sources aggregated into the FinGPT sentiment corpus (alongside FiQA-SA, Twitter Financial News Sentiment, and News-With-GPT-Instructions), so overlap is expected by construction. We measured it directly.

Procedure: We normalize both corpora (lowercasing, Unicode NFKC folding, punctuation stripping, whitespace collapse, which is essential because Financial PhraseBank uses spaced punctuation such as "USD 2.3 mn .") and test each evaluation sentence for (a) exact and (b) normalized membership in the training inputs, recording the training-side gold label for every match. The full script is released as leakage_analysis.py / measure_leakage_local.py.

Findings: The overlap is severe and is summarized in Table 9: 1,699 of the 2,259 unique evaluation sentences (75.2 %) appear verbatim in the training inputs, and in 100 % of those matches the training-side gold label is identical to the Financial PhraseBank label, uniformly across classes. Reproducing the training script's own train test split(test_size=0.05, seed=42), ≈ 1,614 of these leaked sentences fall in the 95 % fine-tuning partition. The model was therefore scored on the exact (sentence, label) pairs it was trained on, which fully accounts for the 99.56 % headline accuracy.

Near-duplicate audit: Exact and normalized matching cannot catch paraphrases or lightly edited variants, so we additionally computed, for every retained evaluation sentence, its maximum token-set Jaccard similarity against all 30,209 unique normalized training inputs (inverted-index blocked; released as near_duplicate_audit.py). The near-duplicate tail is small

Table 8: Corrected results on exact-match-decontaminated evaluation sets under the fixed evaluation harness (eval_harness_fixed.py + stats_report.py): the decontaminated 560-sentence FPB remainder, FiQA-SA (all splits pooled then row-wise decontaminated against the FinGPT corpus, n = 235), and Twitter Financial News Sentiment (validation split, row-wise decontaminated, n = 2,373). Decoding is greedy and deterministic (single pass; identical outputs across seeds by construction)

	FPB-560		FiQA-SA		TFNS	
2-3(lr)4-5(lr)6-7 **Model**	**Acc**	**wF1**	**Acc**	**wF1**	**Acc**	**wF1**
Finistral-7B-LoRA (ours)	**0.9893**	**0.9893**	**0.8766**	**0.8833**	**0.7893**	0.7959
Mistral-7B-v0.1 (zero-shot)	0.0036	0.0070	0.0128	0.0252	0.0013	0.0025
FinGPT-mt-Llama2-7B-LoRA	0.9857	0.9857	0.8085	0.8301	0.7644	0.7713
FinGPT-Llama-3-8B-LoRA	0.9339	0.9410	0.6213	0.7219	0.7804	**0.8211**
FinGPT-Falcon-7B-LoRA	0.9696	0.9697	0.8553	0.8594	0.7307	0.7377
FinGPT-Bloom-7B1-LoRA	0.8946	0.8934	0.6936	0.6914	0.6827	0.6986
FinBERT (ProsusAI)	0.9643	0.9648	0.5149	0.6147	0.7252	0.7329

***Bold** marks the best value per column. Complete statistics (macro F1, unparseable rates, McNemar p-values, and bootstrap confidence intervals for every comparison) are in Appendix A; all comparisons are conditional on each model's native prompt template (Section 5.5 and the ablation show template choice alone can shift accuracy by 9–10 points).*

but nonzero: on FPB-560, 22 sentences (3.9 %) have 119 (5.0 %) reach $J \geq 0.7$ and 15 (0.6 %) $J \geq 0.9$. Even under the worst-case assumption that every $J \geq 0.7$ sentence is answered by memorisation, Finistral's FPB-560 accuracy changes by at most 3.9 points, so near-duplicate leakage cannot account for the corrected results. We accordingly describe our sets as *exact-match-decontaminated* rather than leakage-free, and release the audit alongside the data.

Table 9: Train/test overlap between FinGPT/fingpt-sentiment-train and Financial PhraseBank sentences all agree (locally reproduced).

Quantity	Value
Unique evaluation sentences	2,259
Verbatim (exact) overlap with training	1,699 (75.2 %)
Normalized overlap with training	1,699 (75.2 %)
Label agreement among matched	1,699 / 1,699 (100 %)
Decontaminated remainder (unseen)	560 (24.8 %)

Decontamination and corrected evaluation: We remove every leaked sentence and retain the 560-sentence disjoint remainder (neutral 353, positive 130, negative 77) as an exact-match-decontaminated Financial PhraseBank test set (released as fpb_decontaminated.csv). Because even this remainder shares Financial PhraseBank's stylistic distribution, we additionally evaluate on external sets, decontaminated *row-wise* against the FinGPT training inputs under the same normalization: FiQA-SA (all splits pooled, then leaked rows removed, $n = 235$; pooling is necessary because FiQA is itself a FinGPT constituent and its test split alone is 78 % contaminated) and the Twitter Financial News Sentiment validation split ($n = 2{,}373$ after removing 15 leaked/duplicate rows). Dataset provenance, label mappings, and dropped-row counts are released in data_eval/PROVENANCE.md. Corrected numbers, with per-example McNemar tests and paired-bootstrap 95 % confidence intervals against every baseline, are reported in Table 8.

Harness defects underlying the baselines: Independently of contamination, the original baseline scores (13–23 %, below the 61.6 % majority-class floor) are artifacts of the evaluation harness, not of the models. The committed notebook logs contain hundreds of padding_side=right warnings for decoder-only models; generation was called as generate (**tok, max_length=512) with padding=True (so max_length is ignored and generations ramble); a single Alpaca prompt was applied to every model rather than each adapter's native template; a quantization config built from invalid BitsAndBytesConfig keyword arguments silently degraded to plain int8; and the label parser defaulted unmatched outputs to "neutral." The corrected harness (eval_harness_fixed.py) fixes each of these and reports per-baseline confusion matrices.

6. Result and discussion

6.1 Quantitative gains

The large apparent margins reported in our original submission (+69 F1 over zero-shot Mistral-7B; +78–96 F1 over the FinGPT LoRA baselines; +3–6 accuracy points over FinBERT) do not survive scrutiny and are withdrawn. As Section 5.5 establishes, the Finistral side of every such comparison was inflated by 75.2 % train/test overlap, while the FinGPT baselines were deflated below their majority-class floor by harness defects. The near-perfect per-class F1 on the contaminated split (Negative 98.7 %, Neutral 99.4 %, Positive 99.3 %) and the reduction of errors to "10 of 2,264" are likewise consequences of memorization, not generalization, and the confusion matrix in Fig. 5 should be read as a diagnostic of contamination rather than of model quality. The honest performance picture is given by the decontaminated and external evaluations in Table 8. On the decontaminated FPB remainder, Finistral attains 98.93 % accuracy with per-class F1 of 0.974 (negative), 0.993 (neutral), and 0.989 (positive), but this is *not* a statistically significant advantage over the strongest baseline (FinGPT-mt-Llama2, 98.57 %; McNemar $p = 0.77$), and both models plausibly benefit from FPB's narrow stylistic distribution even after sentence-level decontamination. The external sets are more informative. On FiQA-SA, Finistral leads on accuracy and weighted F1 (87.66 % / 0.8833), significantly above FinGPT-mt-Llama2 ($p = 0.012$, unadjusted), FinGPT-Llama-3 ($p = 7.2 \times 10^{-11}$), and FinBERT ($p = 3.1 \times 10^{-17}$), though not above FinGPT-Falcon (85.53 %, $p = 0.42$); on macro F1, however, Finistral (0.700) *trails* FinGPT-Falcon (0.723), reflecting weakness on FiQA's tiny neutral class (F1 0.276 on 12 examples). Appendix A reports macro F1 for every comparison. On TFNS, Finistral (78.93 %) is statistically tied with the strongest baseline, FinGPT-Llama-3 (78.04 %, $p = 0.41$), and significantly above the remaining adapters and FinBERT (mt-Llama2 $p = 2.1 \times 10^{-3}$; Falcon, Bloom, and FinBERT all $p < 10^{-8}$). The zero-shot Mistral-7B backbone emits almost no parseable label under strict parsing (99 % unparseable outputs): fine-tuning confers, at minimum, output-format compliance and, given the fine-tuned models' accuracy, task competence with it. All p-values are unadjusted across the 21 pairwise tests (Appendix A states which conclusions survive a Holm–Bonferroni correction); notably, *no comparison against the strongest per-dataset baseline reaches significance* (FPB-560 $p = 0.77$; FiQA-SA vs Falcon $p = 0.42$; TFNS vs Llama-3 $p = 0.41$). In sum: the corrected evaluation shows Finistral statistically indistinguishable from the best FinGPT-class adapter on each dataset, significantly better than several others, and clearly ahead of FinBERT off-distribution, not the categorical superiority the original submission asserted.

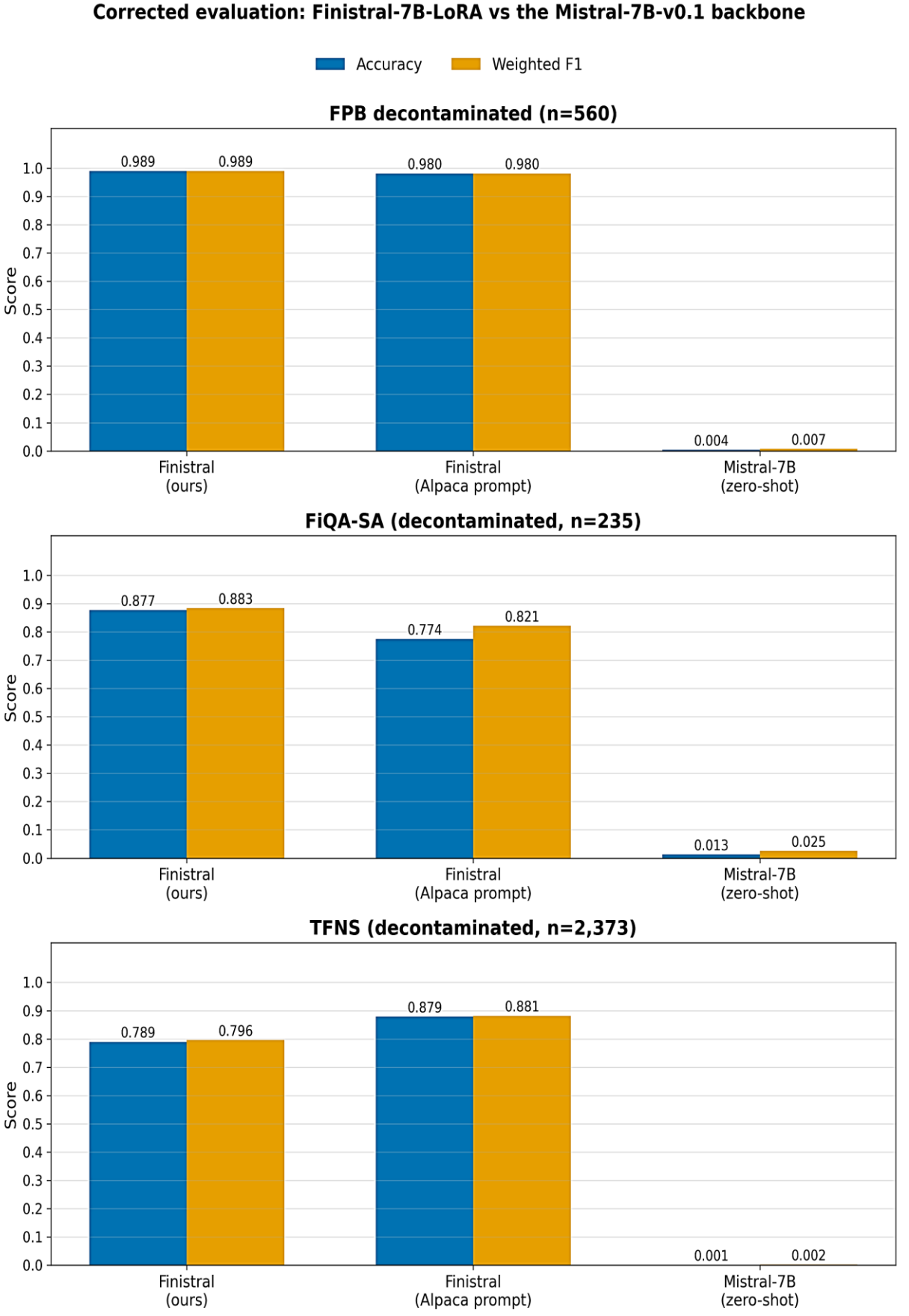

Fig. 3: Corrected comparison on the exact-match-decontaminated sets: Finistral-7B-LoRA (training template and Alpaca-template ablation) vs the Mistral-7B-v0.1 zero-shot backbone on the decontaminated FPB-560, FiQA-SA (n = 235), and TFNS (n = 2,373) sets.

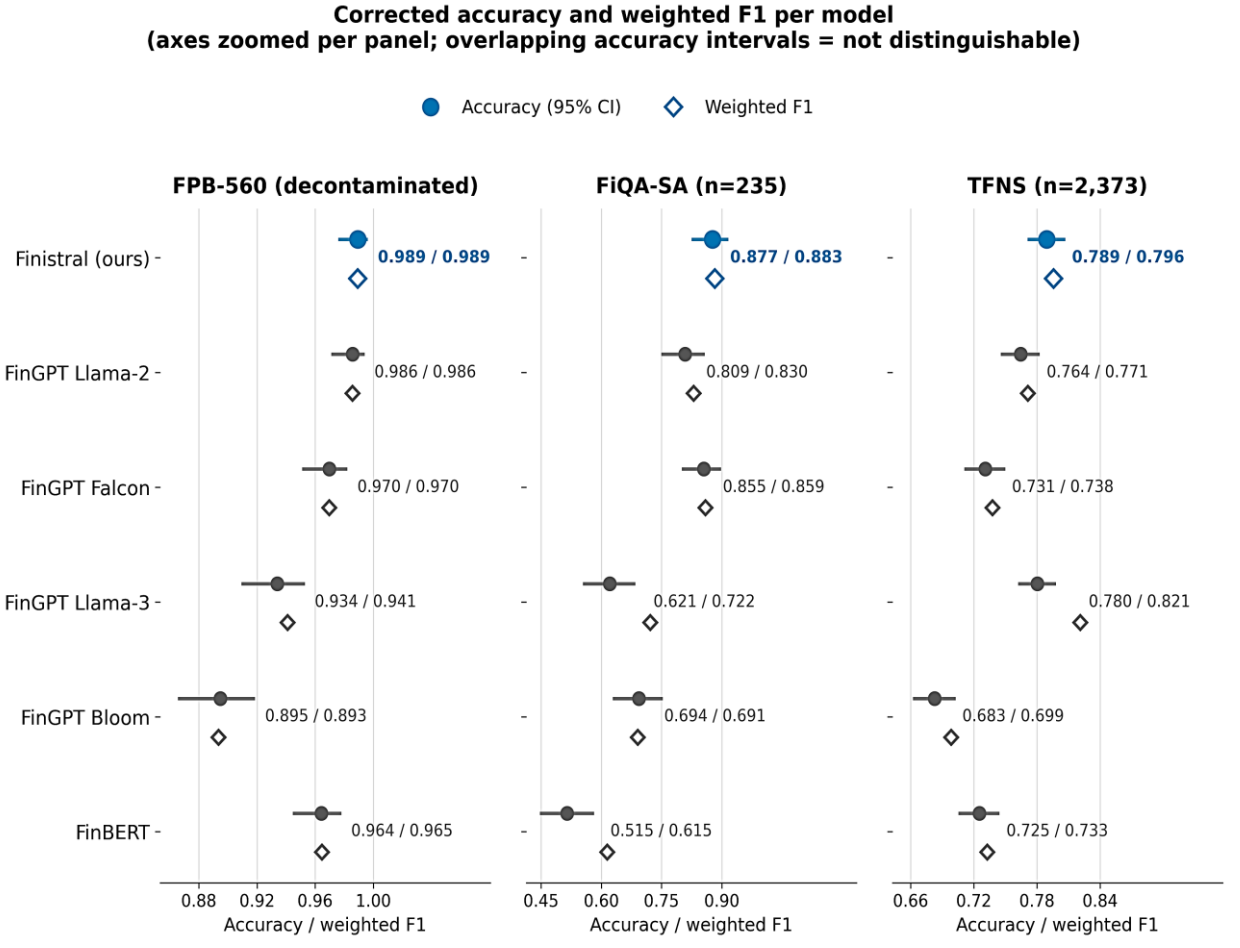

Fig. 4: Corrected accuracy (filled circles, with 95 % Wilson confidence intervals) and weighted F1 (open diamonds) for Finistral-7B-LoRA against all FinGPT adapter baselines and FinBERT on the exact-match-decontaminated sets; numeric labels give accuracy / weighted F1. Axes are zoomed per panel to resolve small differences; because value is encoded by position rather than bar length, the non-zero origin introduces no visual exaggeration. Overlapping accuracy intervals indicate differences that are not statistically distinguishable. Note the overlap with FinGPT-mt-Llama2 on FPB-560, FinGPT-Falcon on FiQA-SA, and FinGPT-Llama-3 on TFNS, matching the McNemar results in Appendix A. The two metrics do not always agree: on TFNS, FinGPT-Llama-3 attains a higher weighted F1 (0.821) than Finistral (0.796) despite lower accuracy, and on FiQA-SA FinGPT-Falcon leads on macro F1 (Appendix A). The unparseable-dominated zero-shot backbone is omitted here and shown in Fig. 3.

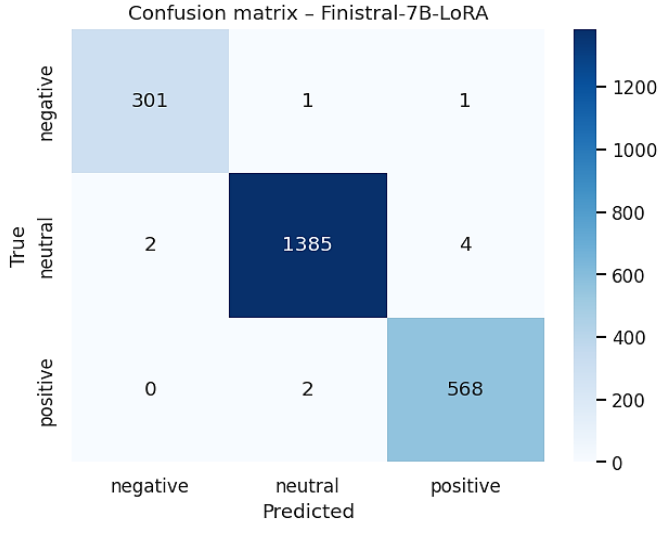

Fig. 5: Confusion matrix of Finistral-7B-LoRA on the original, contaminated full split, retained solely as a contamination diagnostic (Section 5.5); the corrected per-model confusion matrices appear in Fig. 6.

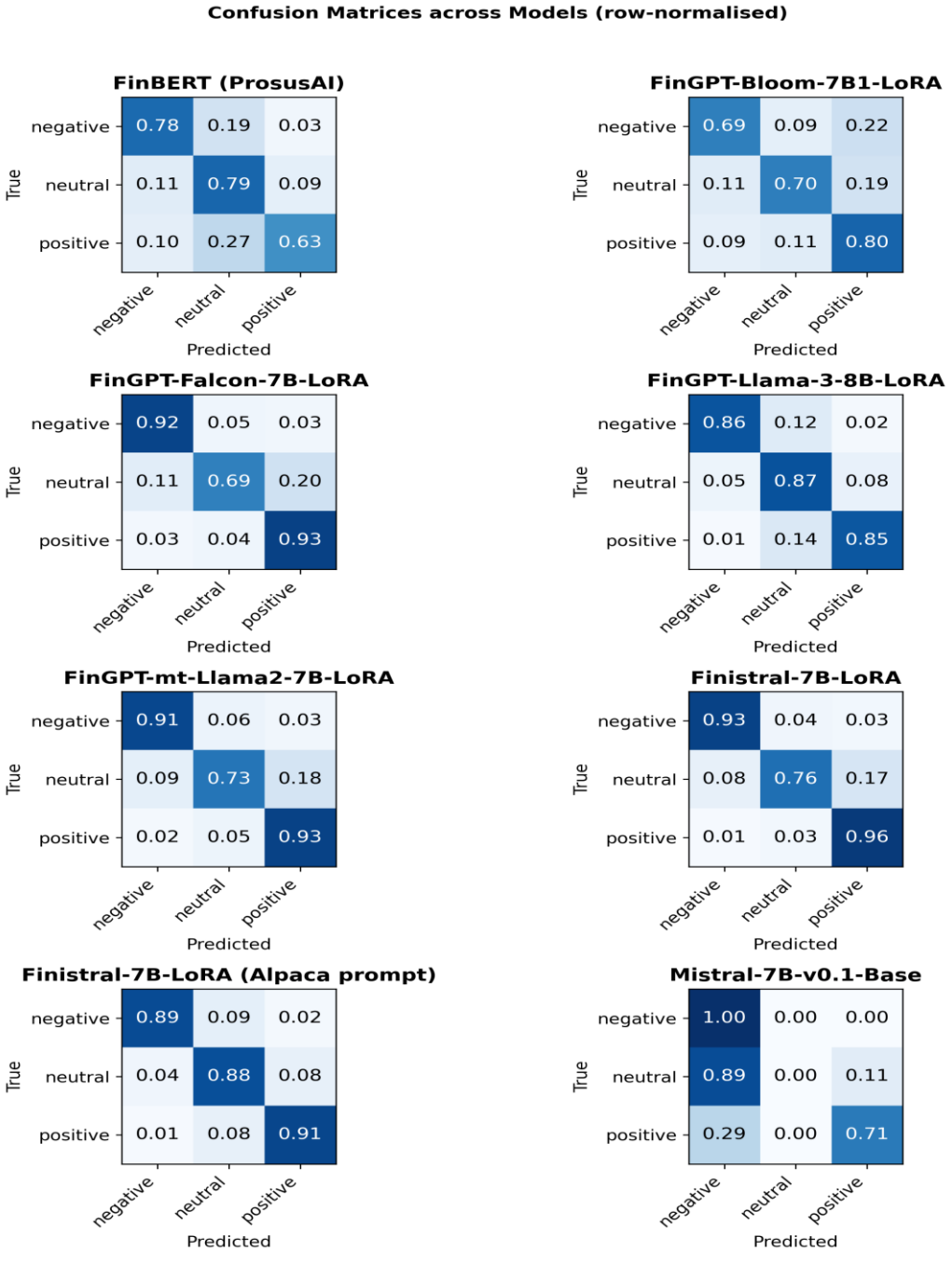

Fig. 6: Row-normalized confusion matrices for every evaluated model under the corrected protocol on the exact-match-decontaminated sets (all three pooled; per-dataset and raw-count variants are released with the code). The unparseable column shows strict-parser failures, dominant only for the zero-shot backbone.

Table 10: Efficiency vs. Quality Tradeoffs

Dimension	Finistral-7B-LoRA	Best FinGPT adapter
Trainable params	41.94 M ($r = 16$, all-linear)	2.4–6.3 M ($r = 8$)
Adapter size	83.9 MB (fp16)	5–14 MB
Train time (2 ×A100)	≈ 120 min	(published adapters)
Weighted F1 (FPB-560 / FiQA / TFNS)	0.989 / 0.883 / 0.796	0.986 / 0.859 / 0.821†

†*Best FinGPT adapter per dataset: FinGPT-mt-Llama2 on FPB-560, FinGPT-Falcon on FiQA-SA, FinGPT-Llama-3 on TFNS.*

6.2 Training dynamics insight

Validation loss reached its minimum (0.1009) at epoch 2 and then rose, a classic sign of incipient overfitting. Selecting the epoch-2 checkpoint therefore maximizes generalization

6.3 Efficiency vs. quality tradeoffs

The original version of this table juxtaposed Finistral's

contaminated 0.996 F1 against a FinGPT baseline's harness-broken 0.211 F1; both numbers were invalid for comparison and have been replaced with corrected figures from the exact-match-decontaminated evaluation. Two honest observations follow. First, the efficiency columns (parameter count, adapter size, train time) are unaffected by contamination and are absolute virtues of the recipe: a competitive financial-sentiment adapter for well under one GPU-day, on academic infrastructure, end-to-end reproducible. Second, however, Finistral is *not* the most parameter-efficient model in its own comparison: FinGPT-mt-Llama2 trains 6.7 × fewer parameters (6.29 M vs 41.94 M) yet is statistically indistinguishable from Finistral on FPB-560 ($p = 0.77$) and competitive on TFNS, with Finistral's clearest win confined to FiQA-SA. The defensible claim is therefore accessibility and reproducibility of the full recipe and evaluation stack, not parameter-count superiority. Identifying the minimal adapter capacity at which this performance saturates is exactly what the deferred rank/target-module ablation (released as ablation_and_seeds.py) would settle.

6.4 Prompt-template ablation

Because the original evaluation applied a single Alpaca-style prompt to every model, mismatched to Finistral's own [INST] training template (Section 3.3), we quantify the template effect directly: the identical released adapter evaluated under both templates on all three exact-match-decontaminated datasets. The effect is material but *dataset-dependent*: on FPB-560 the training-matched template is slightly but not significantly better (98.93 % vs 98.04 %; McNemar $p = 0.18$); on FiQA-SA it is decisively better (87.66 % vs 77.45 %, +10.2 points, $p = 1.9 \times 10^{-4}$); yet on TFNS the *mismatched* Alpaca template wins (87.86 % vs 78.93 %, $p < 10^{-32}$), plausibly because TFNS's informal tweet register sits closer to the generic instruction framing than to the financial-news style dominant in the fine-tuning corpus. Two conclusions follow, and we state both: evaluation protocols must *control and disclose* the prompt template, since a 9–10-point swing dwarfs most model-to-model differences in Table 8; and a training-matched template is a sensible default but not a universally optimal one. We report Finistral's headline numbers under its training template throughout, which on TFNS is the *conservative* choice.

6.5 Error analysis

Across the three exact-match-decontaminated evaluations, Finistral misclassifies 535 sentences (6 on FPB-560, 29 on FiQA-SA, 500 on TFNS); we tag each with rule-based failure-mode categories (released with the per-example predictions). The distribution: numeric/guidance constructions 187, negation 48, sarcasm or irony 32, conditional or hedged statements 21, entity confusion 14, mixed sentiment 2, and 231 uncategorised. Three recurring patterns dominate. *Cost framings of positive events*: "RBS Pays $1.7 Billion to Scrap U.K. Treasury's Dividend Rights" (gold neutral, predicted negative): a large outflow number pulls the prediction negative even when the event is strategically neutral. *Negated or self-correcting market language*: "Barclays Bonds Rise as Lender Cuts Dividends to Shore Up Capital" (gold positive, predicted negative): the model anchors on "cuts dividends" and misses that the sentence reports the market's positive reaction. *Hedged speculation*: "Italy may be willing to compromise…" (gold neutral, predicted positive): modal constructions are read as realised outcomes. TFNS contributes most errors both because it is the largest set and because its informal, elliptical tweet register (tickers, retweet fragments, truncated URLs) is furthest from the fine-tuning corpus's newswire style, consistent with the per-class pattern that Finistral over-predicts the polar classes on TFNS (negative recall 0.93 but precision 0.68; positive recall 0.96 but precision 0.59), trading neutral precision for polar recall.

6.6 Deployment footprint

GPU inference: measured single-sentence latency (batch size 1, greedy decoding, max_new_tokens=8, bfloat16, 50 timed generations after 5 warm-ups) is 65.5 ms median / 67.4 ms p95 on an NVIDIA H100; comparable A100-class hardware sits well within real-time budgets for news monitoring. The CPU / 4-bit-GGML latency figure ("78 ms, M1 Pro") from the original submission is withdrawn as implausible for a 7B-scale model and is not re-asserted without measurement (Section "Inference & Deployment").

6.7 Limitations & ethical considerations

Label noise: FinGPT relies on distant supervision; some training sentences may be mislabelled, potentially propagating bias.

Domain scope: The model is English-only and tuned on equity-centric text; performance on commodities or multilingual filings is untested.

Market impact: Releasing accurate sentiment models could amplify herding behaviour if widely adopted.

Evaluation validity / contamination: Our headline in-distribution score was invalidated by 75.2 % train/test overlap (Section 5.5). Even the decontaminated remainder shares Financial PhraseBank's narrow stylistic and topical distribution (Finnish-company financial news), so the external-dataset results should be weighted most heavily; broader, contamination-free benchmarking remains future work.

Single training run: All reported results derive from one fine-tuning run (seed 42). Evaluation itself is deterministic (greedy decoding; identical outputs across

evaluation seeds by construction), but variance across independent *training* runs was not measured; the released ablation_and_seeds.py implements the multi-seed retraining protocol for this study.

Overconfidence: High accuracy on short, single-sentence headlines does not imply robustness to longer documents or adversarial phrasing; calibration and out-of-distribution testing should be explored.

6.8 Key takeaways

A LoRA recipe over Mistral-7B trains a competitive financial-sentiment adapter with negligible compute (≈ 41.94 M trainable parameters, 0.58 % of the backbone, single node), and this efficiency is the work's robust, contamination-independent contribution. Benchmark overlap must be measured, not assumed: 75.2 % of our evaluation set was present in training, and reporting on the contaminated split produced a misleading near-perfect score. Leakage-aware evaluation (exact-match-decontaminated remainder + external datasets, with a released near-duplicate audit) is essential for any model trained on aggregated corpora such as FinGPT.

Evaluation-harness hygiene matters as much as modelling: left-padding, bounded decoding, per-model prompts, and a strict parser change baseline scores by tens of points and are prerequisites for fair comparison.

Training beyond the validation-loss trough yields no benefit and can harm generalization.

7. Future work

Streaming EDGAR & real-time news: Implement continual learning to keep the model current as market language evolves, aligning with FinNLP shared-task goals.

Human-in-the-loop curation: Use active learning loops to clean noisy FinGPT labels and focus annotation on sarcasm or mixed-tone edge cases revealed in our error analysis.

4-bit & QLoRA fusion: Combine QLoRA-style 4-bit NF4 fine-tuning with post-training GGUF/GGML quantisation of the merged model for compact deployments on laptops and edge devices.[6]

GRPO-based reasoning integration: Apply Group Relative Policy Optimization (GRPO) to inject chain-of-thought reasoning into Finistral-7B-LoRA, enabling the model to articulate its sentiment rationale rather than returning a bare label.[18] GRPO eliminates the need for a separate value model, making reinforcement-learning-based alignment feasible within academic compute budgets.

On-device small language models: Distil the reasoning-augmented adapter into sub-3B parameter small language models (SLMs) optimized for on-device inference on consumer hardware such as iPhone 15, Android smartphones, and low-end laptops. Combining GRPO-trained reasoning with aggressive quantization (4-bit/2-bit) and architecture pruning could yield models that run entirely on-device with sub-200 ms latency, eliminating cloud dependency and preserving user privacy for personal finance applications.

8. Conclusion

Finistral-7B-LoRA demonstrates that a financial-sentiment adapter over the compute-efficient Mistral-7B backbone can be trained with only ≈ 0.58 % trainable parameters (≈ 41.94 M), one node of GPU time, and an 84 MB adapter file, entirely within academic compute budgets on Indiana University's Big Red 200 supercomputer. Equally important, this paper documents a cautionary result: the 99.56 % accuracy reported in our original submission was a data-contamination artifact, with 75.2 % of the Financial PhraseBank evaluation set (and its labels) present in the FinGPT training corpus, while the seemingly weak baselines were products of a broken evaluation harness. After decontaminating the test set, correcting the harness, and validating on external datasets, we present an honest, contamination-audited assessment of the recipe (Table 8). We therefore offer two contributions of lasting value: an efficient, fully reproducible PEFT recipe for financial sentiment, and concrete evidence, with released tooling, that contamination auditing and harness hygiene are prerequisites for credible evaluation of LLMs trained on aggregated financial corpora.

Acknowledgments

The authors acknowledge Indiana University's Big Red 200 supercomputing facility for providing the GPU resources used in this study.

CRediT Author Contribution Statement

Ayan Javeed Shaikh: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original draft, Writing – Review & editing. **Fazal Jalil Parkar:** Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft, Writing – Review & Editing. **Zainab Mirza:** Supervision, Writing – Review & Editing. Ayan Javeed Shaikh and Fazal Jalil Parkar contributed equally to the technical work reported in this study, with Ayan Javeed Shaikh acting as lead author. Zainab Mirza contributed in a supervisory and mentoring capacity. All authors have read and agreed to the published version of the manuscript.

Funding declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement

All data supporting this study are publicly available: the trained adapter weights and tokeniser files at https://huggingface.co/Ayansk11/Finistral-7B_lora; the frozen evaluation sets with provenance records (data_eval/), per-example predictions, complete statistics, decontamination and near-duplicate audit scripts, the corrected evaluation harness, training script, and this manuscript's LaTeX source at https://github.com/ayansk11/FinistralAI_code. The underlying corpora (FinGPT/fingpt-sentiment-train, Financial PhraseBank, FiQA-SA, Twitter Financial News Sentiment) are public datasets available from their original distributors. The models and datasets used in this study are publicly available at https://huggingface.co/FinGPT.

Conflict of interest

There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure

The authors declare that no artificial intelligence (AI)-assisted tools were used in the preparation of this work. All technical content, experimental implementation, results, analysis, and interpretations were independently developed, conducted, and verified by the authors.

Supporting Information

Inference scripts and the Gradio GUI are included in the repositories listed in the Data Availability Statement.

Appendix A: Complete Per-Comparison Statistics

Table A1 reproduces the complete statistical record behind Table 8 (released as results_fixed/stats_summary.csv): accuracy, weighted F1, macro F1, the strict-parser unparseable rate, and, for every baseline, the McNemar p-value and paired-bootstrap 95 % confidence interval on the accuracy difference versus Finistral-7B-LoRA (positive = Finistral ahead). Test protocol: exact binomial McNemar when the discordant-pair count $b + c < 25$, continuity-corrected χ^2 otherwise; paired percentile bootstrap with 10,000 resamples. All p-values are unadjusted; with 21 pairwise comparisons, readers applying a Holm–Bonferroni correction should note that the FiQA-SA advantage over FinGPT-mt-Llama2 ($p = 0.012$) does not survive correction, while all comparisons at $p \leq 2 \times 10^{-3}$ do.

Table A1: Complete corrected-evaluation statistics. ΔAcc CI is the paired-bootstrap 95 % interval on (Finistral − baseline) accuracy

Dataset	Model	Acc	wF1	mF1	Unpars.	McNemar p	ΔAcc 95 % CI
FPB-560	Finistral-7B-LoRA	0.9893	0.9893	0.9853	0.000	n/a	n/a
FPB-560	FinGPT-mt-Llama2-7B-LoRA	0.9857	0.9857	0.9798	0.000	0.774	[−0.009, +0.016]
FPB-560	Finistral (Alpaca prompt)	0.9804	0.9804	0.9755	0.000	0.18	[−0.002, +0.020]
FPB-560	FinGPT-Falcon-7B-LoRA	0.9696	0.9697	0.9633	0.000	0.007	[+0.007, +0.034]
FPB-560	FinBERT (ProsusAI)	0.9643	0.9648	0.9518	0.000	0.007	[+0.009, +0.043]
FPB-560	FinGPT-Llama-3-8B-LoRA	0.9339	0.9410	0.9265	0.018	8.1×10^{-7}	[+0.036, +0.077]
FPB-560	FinGPT-Bloom-7B1-LoRA	0.8946	0.8934	0.8517	0.000	5.7×10^{-11}	[+0.068, +0.121]
FPB-560	Mistral-7B-v0.1 (zero-shot)	0.0036	0.0070	0.0135	0.995	1.3×10^{-121}	[+0.975, +0.995]
FiQA-SA	Finistral-7B-LoRA	0.8766	0.8833	0.6999	0.000	n/a	n/a
FiQA-SA	FinGPT-Falcon-7B-LoRA	0.8553	0.8594	0.7226	0.000	0.424	[−0.021, +0.064]
FiQA-SA	FinGPT-mt-Llama2-7B-LoRA	0.8085	0.8301	0.6743	0.000	0.012	[+0.021, +0.119]
FiQA-SA	Finistral (Alpaca prompt)	0.7745	0.8209	0.6525	0.000	1.9×10^{-4}	[+0.055, +0.153]
FiQA-SA	FinGPT-Bloom-7B1-LoRA	0.6936	0.6914	0.5283	0.000	2.9×10^{-7}	[+0.119, +0.247]
FiQA-SA	FinGPT-Llama-3-8B-LoRA	0.6213	0.7219	0.5795	0.072	7.2×10^{-11}	[+0.187, +0.323]
FiQA-SA	FinBERT (ProsusAI)	0.5149	0.6147	0.4726	0.000	3.1×10^{-17}	[+0.294, +0.430]
FiQA-SA	Mistral-7B-v0.1 (zero-shot)	0.0128	0.0252	0.0176	0.987	1.3×10^{-45}	[+0.817, +0.906]
TFNS	Finistral (Alpaca prompt)	0.8786	0.8811	0.8598	0.000	1.7×10^{-33}	[−0.104, −0.075]
TFNS	FinGPT-Llama-3-8B-LoRA	0.7804	0.8211	0.7906	0.093	0.407	[−0.011, +0.029]
TFNS	Finistral-7B-LoRA	0.7893	0.7959	0.7776	0.000	n/a	n/a
TFNS	FinGPT-mt-Llama2-7B-LoRA	0.7644	0.7713	0.7527	0.000	0.002	[+0.010, +0.040]
TFNS	FinGPT-Falcon-7B-LoRA	0.7307	0.7377	0.7223	0.000	2.4×10^{-11}	[+0.042, +0.075]
TFNS	FinBERT (ProsusAI)	0.7252	0.7329	0.6679	0.000	9.5×10^{-9}	[+0.043, +0.086]
TFNS	FinGPT-Bloom-7B1-LoRA	0.6827	0.6986	0.6449	0.000	1.2×10^{-21}	[+0.085, +0.128]
TFNS	Mistral-7B-v0.1 (zero-shot)	0.0013	0.0025	0.0047	0.995	$< 10^{-300}$	[+0.772, +0.804]

Appendix B: Corrections Relative to the Original Submission

For archival transparency, Table B1 consolidates every substantive correction made during revision; the full point-by-point account is in the Response to Reviewers and the repository changelog.

Table B1: Corrections relative to the original submission

Original claim	Defect	Corrected (location)
99.56 % accuracy, state of the art	75.2 % train/test overlap	Exact-match-decontaminated + external evaluation (Table 8)
+78 F1 over FinGPT baselines	Broken evaluation harness	Baselines recover to 0.89–0.99 on FPB-560 (Sec. 5.5)
Trained with Unsloth, "8-bit NF4"	Neither used in released code	Plain transformers+peft, bf16 (Sec. 4.1)
LoRA on q/v only; ≈9 M (0.2 %) params	Neither matches the released adapter	All seven linear modules; 41.94 M (0.58 %), verified from artifact file sizes (Sec. 4.2)
LoRA dropout 0.10	Training config says 0.05	0.05 (Table 3)
168 MB "4-bit" GGML adapter	File size implies fp32	fp32 GGML export, 167.8 MB (Sec. 4.4)
78 ms CPU latency (M1 Pro)	No supporting benchmark retained	Withdrawn; measured GPU latency reported instead (Sec. 6.6)
Baseline backbone unsloth/mistral-7b-v0.2	Inconsistent with Finistral's v0.1 backbone	All runs on mistralai/Mistral-7B-v0.1 (Sec. 5.1)
Eval prompt = training prompt	Alpaca prompt used at eval, [INST] at training	Aligned; mismatch quantified in the ablation (Sec. 6.4)
Val loss 0.1008 vs 0.1009	Inconsistent reporting	0.1009 (Table 5)

References

1 T. Loughran, B. McDonald, When is a liability not a liability? textual analysis, dictionaries, and 10-Ks, *The Journal of Finance*, 2011, **66**, 35-65, doi: 10.1111/j.1540-6261.2010.01625.x

2 E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of large language models, International Conference on Learning Representations (ICLR), 2021.

3 P. Malo, A. Sinha, P. Korhonen, J. Wallenius, P. Takala, Good Debt or Bad Debt: Detecting semantic orientations in economic texts, *Journal of the Association for Information Science and Technology*, 2014, **65**, 782–796, doi: 10.1002/asi.23062.

4 J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019, 4171–4186, doi: 10.18653/v1/N19-1423

5 D. Araci, FinBERT: Financial sentiment analysis with pre-trained language models, arXiv preprint, 2019, arXiv:1908.10063, doi: 10.48550/arXiv.1908.10063.

6 T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient Finetuning of Quantized LLMs, *Advances in Neural Information Processing Systems* (NeurIPS), 2023, **36**, 10088-10115, doi: 10.52202/075280-0441.

7 Z. Han, C. Gao, J. Liu, J. Zhang, S. Q. Zhang, Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024, arXiv:2403.14608, doi: 10.48550/arXiv.2403.14608.

8 D. Wang, J. Patel, D. Zha, S. Y. Yang, X.-Y. Liu, FinLoRA: Benchmarking LoRA methods for fine-tuning LLMs on financial datasets, arXiv preprint, 2025, arXiv:2505.19819, doi: 10.48550/arXiv.2505.19819.

9 B. Zhang, H. Yang, X. Y. Liu, Instruct-FinGPT: Financial sentiment analysis by instruction tuning of general-purpose large language models, 2023, arXiv:2306.12659, doi: 10.48550/arXiv.2306.12659.

10 H. Yang, X. Y. Liu, C. D. Wang, FinGPT: Open-source financial large language models, 2023 arXiv:2306.06031, doi: 10.48550/arXiv.2306.06031.

11 C. C. Chen, G. I. Winata, S. Rawls, A. Das, H. H. Chen, H. Takamura, Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing, Association for Computational Linguistics, 2023, (FinNLP 2023).

12 Q. Xie, W. Han, Z. Chen, R. Xiang, X. Zhang, Y. He, M. Xiao, D. Li, Y. Dai, D. Feng, Y. Xu, FinBen: A Holistic Financial Benchmark for Large Language Models,

Advances in Neural Information Processing Systems (NeurIPS) Datasets & Benchmarks, 2024, arXiv:2402.12659, doi: 10.52202/079017-3033.
13 Z. Liu, X. Guo, Z. Yang, F. Lou, L. Zeng, J. Niu, M. Li, Q. Qi, Z. Liu, Y. Han, D. Cheng, Fin-R1: A Large Language Model for Financial Reasoning through Reinforcement Learning, 2025, arXiv:2503.16252, doi: 10.48550/arXiv.2503.16252.
14 A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L.R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, W.E. Sayed, Mistral 7B, 2023, arXiv:2310.06825doi: 10.48550/arXiv.2310.06825.
15 Y. Oren, N. Meister, N. Chatterji, F. Ladhak, T.B. Hashimoto, Proving Test Set Contamination in Black Box Language Models, International Conference on Learning Representations (ICLR), 2024, **2024**, 16354-16372.
16 R. Xu, Z. Wang, R.-Z. Fan, P. Liu, Benchmarking benchmark leakage in large language models, 2024, arXiv:2404.18824, doi: 10.48550/arXiv.2404.18824.
17 M. Roberts, H. Thakur, C. Herlihy, C. White, S. Dooley, Data contamination through the lens of time, 2023, arXiv:2310.10628, doi: 10.48550/arXiv.2310.10628.
18 Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y.K. Li, Y. Wu, D. Guo, DeepSeekMath: pushing the limits of mathematical reasoning in open language models, 2024, arXiv:2402.03300, doi: 10.48550/arXiv.2402.03300.