Learning to Replicate Expert Judgment in Financial Tasks

Sarah Su, Kevin Zhu, Emily Xiao, Rohan Alur, Daniel Kang
(Bridgewater AIA Labs) in collaboration with Thinking Machines

Jun 30, 2026



Judging information

Outperforming the market is hard. When every investor has access to the same sources of public information, alpha must come from unique insight built on taste and judgment. A strong investor's judgment is difficult to articulate and teach directly to others, whether human or AI. It comes from experience.

Even when we decompose an investor's job into its simplest constituent tasks, those tasks turn out to be surprisingly difficult for LLMs. In this post, we consider a simple special case: filtering and processing financial documents to surface information relevant to investment decisions.

Investors are bombarded with information every day: news articles, research reports, company documents, emails, internal write-ups, and more. Reading is the easy part. The real work is the small, repeated judgments carried over it—filtering, interpreting, segmenting, and identifying where the useful signal lies. These judgments are embedded throughout an investor's daily workflow and consume substantial time.

We wanted to see if we could automate the information triage task: identifying what is relevant and interesting to read. This alone could greatly augment investors' productivity, letting them spend their freed up attention on higher-level synthesis and decision making.

Given that LLMs perform poorly on simple financial tasks, we asked: is it possible to teach LLMs financial judgement? We find that with **high-quality human annotations**, we can teach LLMs to interpret text with expert-level taste and judgement. **Our proprietary model outperforms all frontier models we tested on information accuracy and recall, at a fraction of their cost.**

We describe our training process and results on a subset of data cleared for public release. Based on our results, we further describe the seeds of a vision of *differentiated intelligence*, with models tuned for specific organizational needs.

Frontier model performance

We evaluated models on six information filtering tasks drawn from investors' daily workflows. Beyond these tasks, we have many others internally that show similar patterns to these six tasks: frontier models we tested on underperform compared to our internally trained models.

We measured accuracy—the percentage of documents that were correctly labeled according to our investors. For classification tasks, we also calculated the F1 score.[1]

01 Financial Article Relevancy

Given a financial article, classify whether it is relevant to a C-suite investment professional.

EVAL METRICS
F1 score, Accuracy

02 Central Bank Document Relevancy

Given a central bank document, classify whether it signals the direction of future interest rate changes.

EVAL METRICS
F1 score, Accuracy

03 Generic Document Relevancy

Given an investor's question and a research document, classify whether the document helps answer it.

EVAL METRICS
F1 score, Accuracy

04 Ad Hoc Content Labeling

Research documents are either recurring (repeated boilerplate) or mixed (boilerplate plus one-off, issue-specific analysis). Classify which, and find the last page of issue-specific content.

EVAL METRICS
Accuracy

05 Document Truncation

Identify where boilerplate content begins in a document.

EVAL METRICS
Exact Match Accuracy

06 Email Truncation

Identify where boilerplate content begins in an email.

EVAL METRICS
Exact Match Accuracy

Figure 1: The six financial tasks we evaluate in this blog post, each drawn from the routine work of an investor.

These tasks are trivial for investors, but they get stuck when articulating their decision process. Consider the following example of classifying a news article as relevant to an investment professional below:

Not relevant

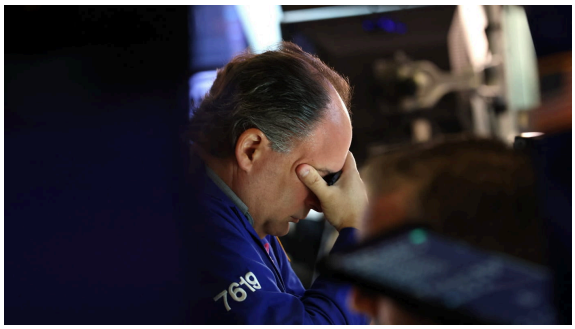
ft.com
Trump insists Greenland is his

Relevant

ft.com
US stocks close sharply lower after Trump threatens new China tariffs



© Jeremy Banx



Biggest one-day drop in S&P 500 since April brings weeks long rally to a halt © AFP/Getty Images

Figure 2: Example of judging the relevance of a financial article to US markets. Source: Financial Times.

The Greenland example is unlikely to be taken seriously given the context of the article, while the China tariffs are highly relevant. Yet both examples touch on geopolitics and finance.

In contrast to our investors, frontier models we tested on perform surprisingly poorly. Variants of Gemini, Claude, and GPT averaged a mere ~50% accuracy when given a prompt that simply states each of the six tasks to perform.

We first tried to improve LLM performance with stronger prompting. Our experts wrote instructions based on real task descriptions, and also suggested reframing certain tasks. For example, while an article about a small IPO is clearly financially relevant, it lacks the broad significance that would make it interesting to a macroeconomic investor at Bridgewater. LLM performance on the article classification task improved when they were asked to sort news stories into three labels: relevant and interesting, relevant but uninteresting, and irrelevant.

These changes boosted their accuracy from a coin flip to the mid-70s. We saw no further gains in accuracy from automatic prompt-optimization methods. With our best prompts the frontier models we tested on still achieved less than 80%

accuracy—the threshold investors expect from a system they could trust in their daily workflow.

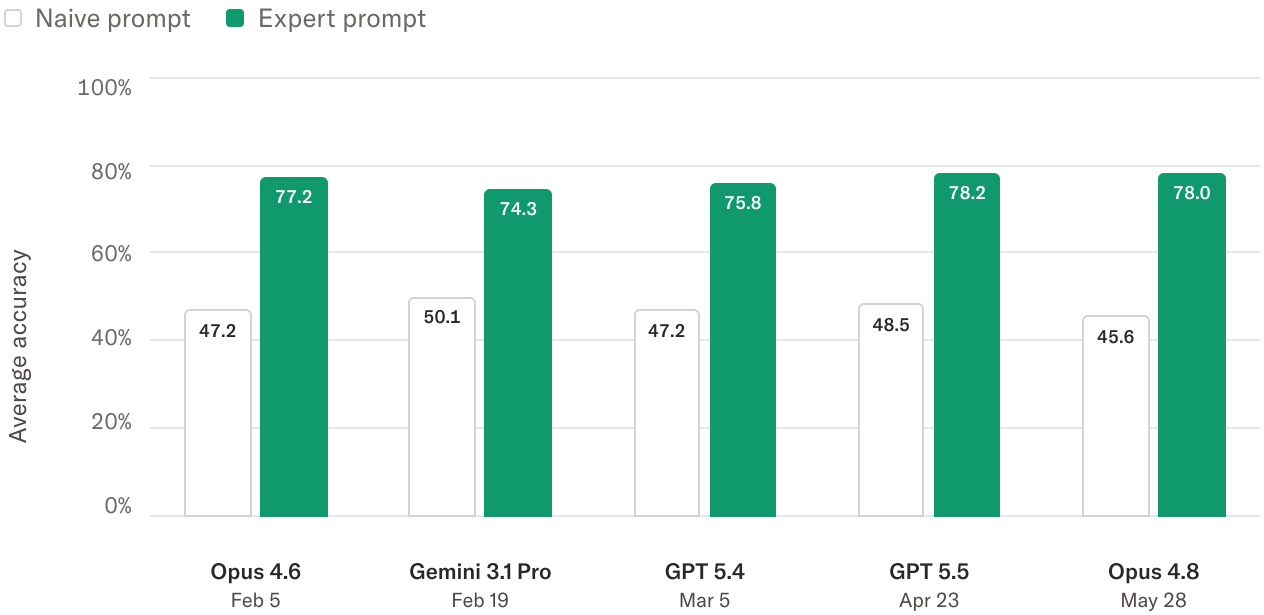


Figure 3: Accuracy & Positive Class F1 score of frontier models on our financial tasks after manual and automatic prompt engineering. F1 score is averaged across our 3 classification tasks, and accuracy is averaged across all 6 tasks.

Our results also suggest that newer models aren't improving rapidly at this task, especially per dollar spent. GPT 5.4 costs 43% more than 5.2 but is only marginally more accurate.

An explicit prompt can only convey the intuition an expert is able to put into words, while the judgments that matter most are often the hardest to articulate. Fine-tuning sidesteps this: rather than contorting the expert's intuition into a static prompt, the training process lets the model develop its own judgment. Could we train open-weight models to outperform frontier models we tested on these tasks?

Training dataset construction

The first challenge of training a custom model was acquiring a dataset that reflects **high-quality investor taste**. In particular, much of the information is only useful

when filtered through an investment professional's judgment.

We initially sourced a dataset from vendors providing non-expert labeling. Models trained on this dataset still performed poorly. After examining the reasoning traces of the model we realized that the labels in the dataset were often wrong. Since expert labelers are costly, we devised a verification scheme that routes only the contested examples to experts.

The scheme worked as follows: we trained a model on the dataset from non-expert labelers, then evaluated it on the same data. Examples where the model's answer differed from the labelers' were sent to our experts for reevaluation—if a model couldn't match an example from its own training set then either the example is genuinely difficult, or the original label was wrong. This procedure was used to clean the training set data; the final evaluation was done on a held out test set.

Training recipe

We trained our models on Tinker from Thinking Machines Lab.[2] Tinker allowed us to iterate quickly without worrying about GPU infrastructure.

We chose Qwen3-235B as the base model as its fine-tuning performance is widely studied in the academic literature.

We began with standard GRPO and importance-sampling loss as a simple, critic-free starting point. This baseline approach resulted in a massive jump in the model performance, but it still fell short of our desired 80% threshold.

Model / Training	Average Accuracy	Average Pos F1
Qwen Base	44.8%	55.24%
Qwen + GRPO	73.48%	88.95%

We make the following modifications to our training recipe to push performance farther:

1. Interleaved batching

For our multi-task training recipe, we compared three batching strategies: training each task sequentially, fully mixing tasks within a batch, and interleaving one batch per task in round-robin order. We found interleaving worked best, improving accuracy by 12.1% over fully mixed batches.

2. CISPO loss with asymmetric clipping

We used CISPO loss with asymmetric clipping[3] to replace the standard importance-sampling loss. Across the loss functions and clipping schemes we tried, this performed best, improving accuracy by 10.1% over the importance-sampling baseline.

3. On-policy distillation with strong teachers

We train with on-policy distillation[4] (OPD), constructing the advantage as follows:

$$r = \text{reward} - \beta \cdot \text{avg}(\text{student_lp} - \text{teacher_lp})$$

$$\text{adv}_i = r_i - \text{avg}(r)$$

The reward is penalized when the student drifts from the teacher's distribution, regularizing the policy while it learns the task.

Every 20 steps, we promote the current checkpoint to the teacher—but only if validation accuracy has reached a new high, so we never distill toward a weaker model. This gave a further 3.1% gain over a frozen base-model teacher.

Results

Finding the optimal training recipe required several iterations of different approaches. Tinker's accessibility allowed us to run fast experiments and refine our approach.

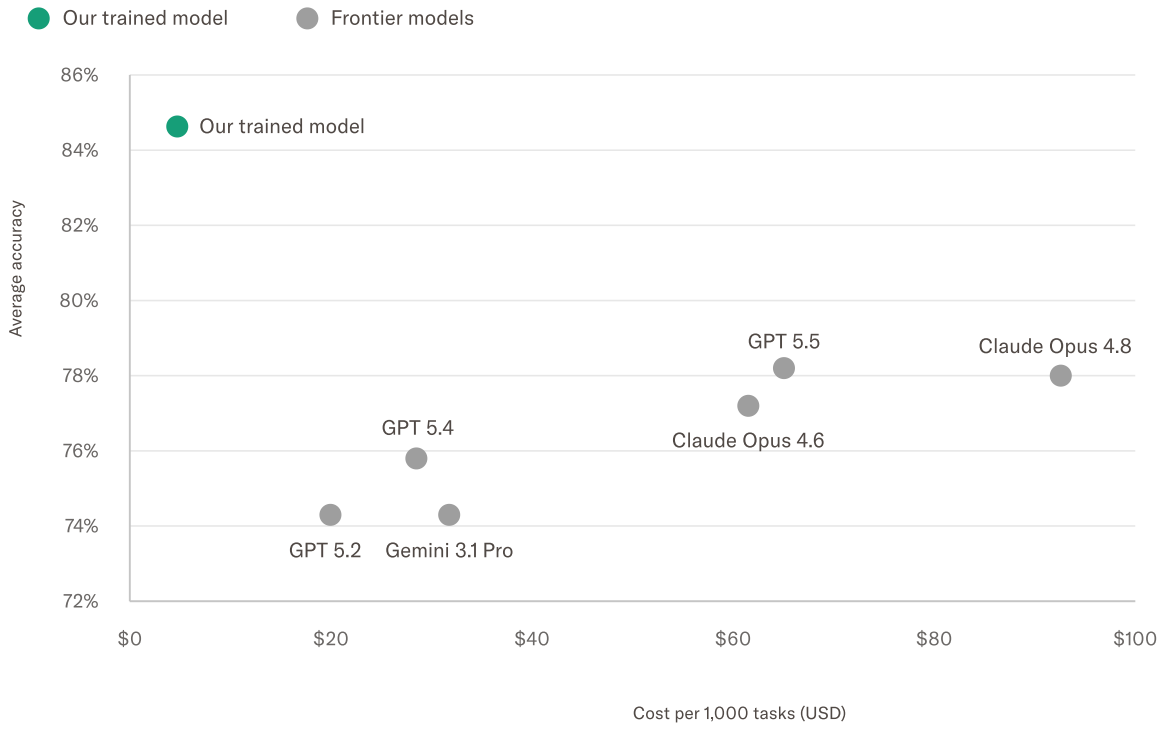


Figure 3: Accuracy versus price for our trained model and frontier models. Our model outperforms frontier models on both dimensions across generations.

Our trained model improves average accuracy from 78.2% to 84.7%, meaning the trained model makes 29.8% fewer mistakes than the best frontier model we evaluated. We find this level of accuracy is sufficient for our daily work.

Our trained model is also vastly cheaper due to its smaller size: a 13.8x reduction in inference costs per task. As we plan to rely on more models trained to help with specific tasks and to scale AI across the organization, cost is an important consideration.

We ablated each part of our training recipe to show how each portion contributes to performance.

Training Method Ablations	Average Accuracy	Avg Pos F1
Qwen + Final Recipe	84.66%	92.99%
⛔ Interleaved Batching	72.18%	89.01%
⛔ CISPO + Asymmetric Clips	74.56%	90.64%
⛔ OPD	72.39%	87.93%
⛔ OPD w/ Best Val Accuracy Teacher	81.55%	89.41%

Figure 4: Each row shows the final recipe with that single component removed (leave one out ablations)

Conclusion

Frontier models we tested on struggle with relatively simple financial tasks, and model advances don't improve performance much. In contrast, we've shown that **high-quality proprietary datasets** labeled by expert investors and used for fine-tuning produce custom models that exceed frontier performance on our tasks. We have found that this outcome holds true well beyond the six tasks we've discussed in this post.

Aside from higher accuracy, custom models are also substantially cheaper. We expect to see more productivity gains from custom model training in the future, especially with the availability of training infrastructure like Tinker that enables rapid experimentation.

Our results show the possibility of a future of differentiated intelligence, where custom models tuned to specific organizational needs outperform frontier models.

Citation

Please cite this work as:

```
Su, Sarah; Zhu, Kevin; Xiao, Emily; Alur, Rohan; Kang, Daniel (Bridgewater A
Thinking Machines Lab: News, June 2026.
```

Or use the BibTeX citation:

```
@article{su2026expertjudgment,
  author = {Sarah Su, Kevin Zhu, Emily Xiao, Rohan Alur, Daniel Kang (Bridge
  title = {Learning to replicate expert judgment in financial tasks},
  journal = {Thinking Machines Lab: News},
  year = {2026},
  note = {https://thinkingmachines.ai/news/learning-to-replicate-expert-judg
}
```

Thinking Machines Lab © 2026