DriMM: Drilling Multimodal Model for Time-Series and Text

Anonymous Authors[1]

Abstract

Multimodal contrastive learning can align time series sensor data with textual descriptions, but its use in industrial settings is still rare. This paper introduces DriMM, a Drilling Multimodal Model that learns joint representations from time series sensor data and textual activity labels from Daily Drilling Reports. DriMM uses large models for time series and pretrained language models to build a shared embedding space across modalities. Our experiments show that DriMM enables cross-modal retrieval and zero-shot classification of drilling activities. As a side effect, the learned mono-modal representations also improve linear probing classification accuracy compared to generic pretrained baselines. These results demonstrate the potential of large models for multimodal learning in domain-specific industrial tasks.

1. Introduction

While recent Large Models for Time Series (LM4TS), such as Chronos (Ansari et al., 2024), Moirai (Woo et al., 2024), and MOMENT (Goswami et al., 2024), generalize effectively across standard benchmarks, their performance declines in specialized domains. (Buiting et al., 2024) have shown that in the Oil & Gas drilling domain, LM4TS are often outperformed by simpler convolutional models. While this was an interesting finding, it is worth noting that in drilling, data from different modalities are inherently captured, allowing us to potentially extend the study of multimodal learning in the era of large models. Drilling operations capture two complementary modalities: (1) time series sensor data that continuously measures physical parameters like hook load, torque, and flow rate; and (2) textual Daily Drilling Reports (DDRs) that provide concise textual summaries of operational activities, written by humans using specialized technical language and abbreviations. To date, no research has explored multimodal learning combining time-series and textual data in the drilling domain. Also, existing multimodal approaches predominantly focus on medical data (Baldenweg et al., 2024; Li et al., 2024), leaving a significant gap in industrial contexts.

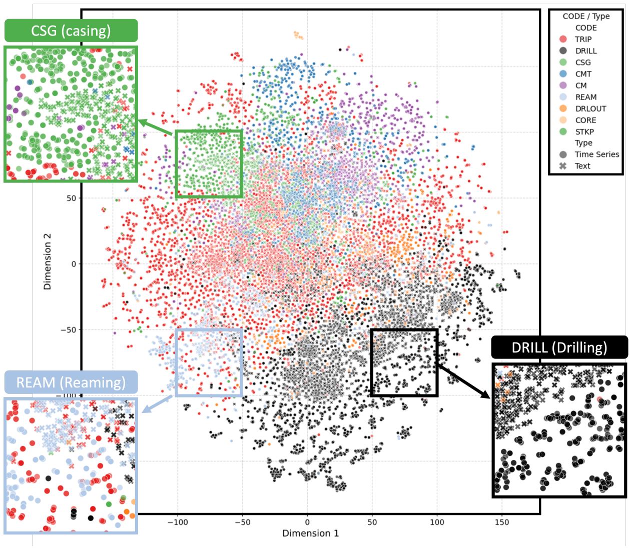


Figure 1. **Joint embedding space learned by DriMM.** Each point is a sensor window (×) or a DDR sentence (○), visualized using PCA and t-SNE. Clusters reflect rig activity classes, showing that DriMM aligns modalities in a meaningful way.

In this paper, we address this gap by studying multimodal learning specifically involving drilling sensor data and DDR text. We trained the first drilling multimodal model called ”DriMM”, on 145 thousand pairs of drilling time-series and their associated drilling textual reports. Technically, we leveraged contrastive learning, utilizing the InfoNCE loss (van den Oord et al., 2018) to align embeddings from sensor data (encoded by LM4TS) and DDR text (encoded by domain-specific models; here RoBERTa). By training DriMM, we enabled the following tasks on drilling data: (1) **Cross-modal retrieval**, allowing queries from one modality to retrieve relevant entries from the other; and (2) **Zero-shot classification** of drilling activities on time-series, based purely on textual prompts.

These capabilities are directly valuable in real-world drilling operations. Indeed, retrieval enables drilling engineers to identify historical patterns given a drilling operation report. Zero-shot classification supports rapid labeling of new data using predefined textual templates, which is especially valuable when labeled datasets are sparse, a common scenario in the drilling domain.

We demonstrate promising results in both tasks. Addition-

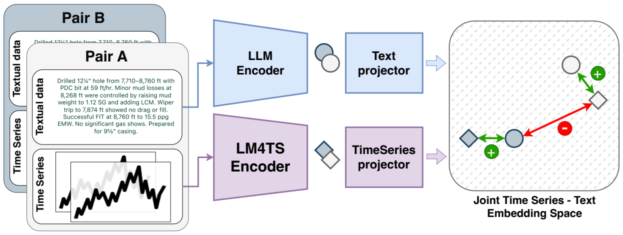

Pair B
Drilled 12¼" hole from 7,710–8,760 ft with
Textual data
Time Series
Pair A
Textual data
Drilled 12¼" hole from 7,710–8,760 ft with PDC bit at 59 ft/hr. Minor mud losses at 8,268 ft were controlled by raising mud weight to 1.12 SG and adding LCM. Wiper trip to 7,874 ft showed no drag or fill. Successful FIT at 8,760 ft to 15.5 ppg EMW. No significant gas shows. Prepared for 9⅝" casing.
Time Series
LLM Encoder
Text projector
LM4TS Encoder
TimeSeries projector
+
+
-
Joint Time Series - Text Embedding Space

Figure 2. **Overview of the DriMM architecture.** The model encodes time series and text pairs using modality-specific encoders: a Large Model for Time Series (LM4TS) and an LLM. Linear projection and normalization map both into a shared embedding space. During training, aligned pairs are pulled together while misaligned ones are pushed apart using a contrastive objective.

ally, we show that multimodal contrastive training enhances the quality of LM4TS embeddings compared to generic pretrained models, improving linear-probing accuracy. Figure 1 illustrates how DriMM organizes sensor and text embeddings in a shared space. The model separates activities like tripping, drilling, and casing, and clusters semantically similar operations across modalities. Rotary drilling states partially overlap due to similar surface signals, but overall, the clusters reflect the physical semantics of operations, supporting the design of our multimodal approach.

2. Methodology

Our objective is to learn a shared embedding space between sensor data and textual descriptions of drilling operations.

2.1. Model Architecture

As illustrated by Figure 2, the model consists of two modality-specific encoders and projection heads:

- **Time Series Encoder:** A pretrained LM4TS (i.e., Moirai or MOMENT) processes windows of multivariate surface sensor data (e.g., hookload, torque, ...). We choose these models because these accept multivariate data.
- **Text Encoder:** A RoBERTa (base) model (Liu et al., 2019) pretrained to the technical language of DDRs.
- **Projection Heads:** Linear layers map both encoders' outputs into a shared embedding space.

2.2. Multimodal Capabilities

Aligning sensor and text modalities in a joint space enables key capabilities that are useful in industrial workflows:

Cross-Modal Retrieval. A signal window can retrieve semantically matching operation descriptions, and vice versa. This supports use cases such as referencing similar past operations, automated search in drilling logs, and quality control without relying on manually curated labels.

Zero-Shot Classification. Domain-specific classes are represented by textual prompts. The text encoder embeds these prompts, which then act as anchors. Sensor signals can be classified by finding the closest prompt in the embedding space. This enables automatic labeling in scenarios where ground truth is unavailable. The anchors/prompts used are provided in Appendix A

Linear Probing. To assess the structure of the learned embeddings, we train a linear classifier using time series embeddings generated by the trained and frozen LM4TS model. Strong linear separability indicates that the model has learned a meaningful representation of drilling activities. This simplifies downstream classification and reduces the need for highly-supervised models.

Together, these capabilities illustrate how contrastive training improves the expressiveness and usability of sensor representations in operational contexts, enabling automation and decision support directly from raw multimodal data.

2.3. Contrastive Learning Objective

We train the model with InfoNCE loss that pulls paired sensor-text embeddings together and pushes unpaired ones apart. Given a batch of N aligned pairs (z_{ts}^i, z_{txt}^i), where z_{ts}^i is the embedding of the i-th time series and z_{txt}^i is the embedding of the corresponding text, the loss is computed in both directions and averaged:

$$L = \frac{1}{2N} \sum_{i=1}^{N} \Bigg[\log \frac{\sum_{j=1}^{N} \exp\left(\text{sim}(z_{ts}^i, z_{txt}^j)/\tau\right)}{\exp\left(\text{sim}(z_{ts}^i, z_{txt}^i)/\tau\right)} + \log \frac{\sum_{j=1}^{N} \exp\left(\text{sim}(z_{txt}^i, z_{ts}^j)/\tau\right)}{\exp\left(\text{sim}(z_{txt}^i, z_{ts}^i)/\tau\right)} \Bigg] \tag{1}$$

$\text{sim}(u, v)$ denotes cosine similarity, τ is a temperature parameter, and N is the number of positive pairs in the batch.

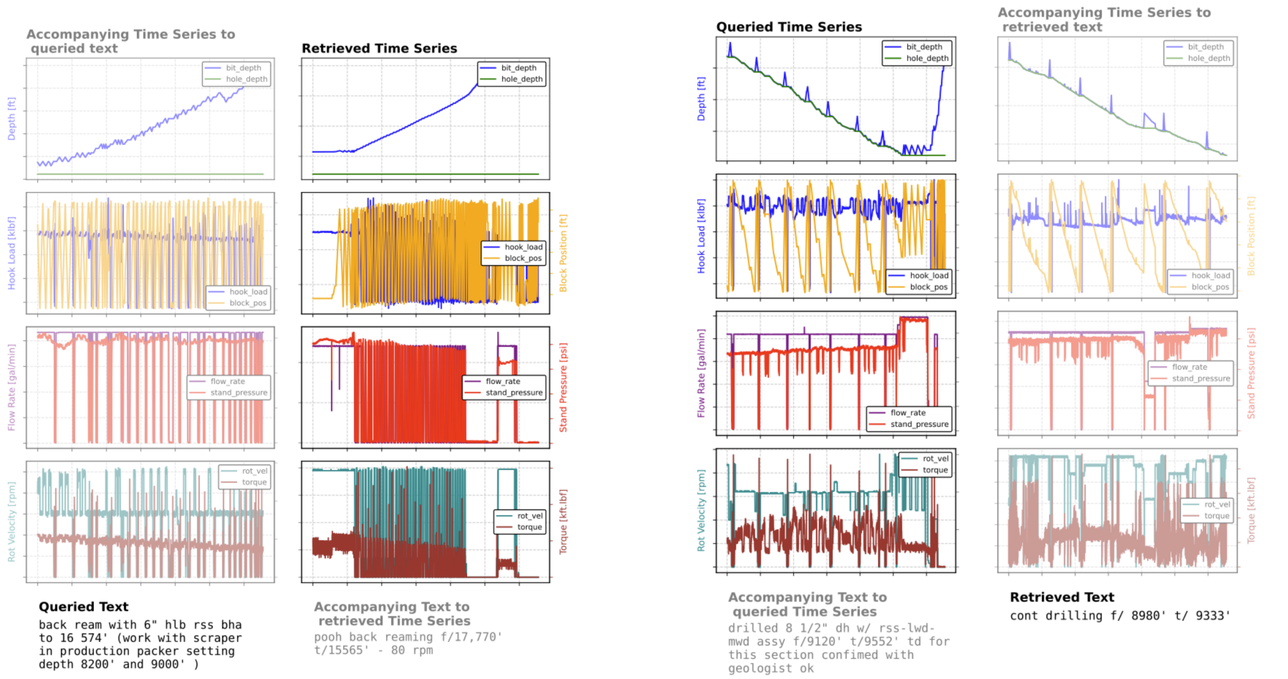
2.4. Training Details

Further details on the training configuration, including optimizer settings, embedding dimensions, and early stopping criteria, are provided in Appendix B.

3. Experimental results

3.1. Data and Operational Activities

The dataset comprises 1787 distinct multivariate time series, each with 10 sensor features (e.g., hookload, torque, flow rate) recorded at a 1-second interval, paired with textual annotations from DDRs. Time series are segmented into 65536-timestep strided windows, subsampled to 512 steps. Each window is associated with a single textual drilling operation e.g., ”RIH 5” BHA to 10,245 ft, tagged top CSG,



(a) **Text to Time Series.** The left column shows the query text and its corresponding time series. The right column shows the top retrieved time series and its corresponding text. *The retrieved time series has the correct class REAM, close depth and sensor values, same activity, and a semantically similar description.*

(b) **Time Series to Text.** The left column shows the query time series and its corresponding text. The right column shows the top retrieved text and its time series. *The model retrieves a text with matching drilling activity and a close semantic description—demonstrating accurate TS-to-text alignment.*

Figure 3. Qualitative results for cross-modal retrieval. (a) shows retrieval from text to time series. (b) same in the opposite direction.

circ @ 10.2 ppg, no losses". This results in 145,715 paired (time series window, text) samples. Understanding the interplay of multiple sensor features is key for interpreting drilling operations.

To evaluate whether the models capture the global semantic structure of drilling operations, we annotate the validation set with high-level activity labels. These labels correspond to nine drilling classes (`DRILL`, `TRIP`, `CSG`, `CM`, `CMT`, `CORE`, `DRLOUT`, `REAM`, `STKP`). They serve as semantic references for interpreting both retrieval and classification performance. The dataset is split 80/20 (train/validation) by distinct time series to prevent information leakage.

3.2. Evaluation Metrics

We report standard metrics tailored to each capability:

Global Retrieval. We use Recall@1, Recall@10, and Recall@100 to quantify how well a query from one modality retrieves relevant entries from the other. Metrics are averaged across both directions (TS→Text and Text→TS).

Class-Level Retrieval. We evaluate retrieval quality at the class level using precision, recall, and F1-score at $k = 1$ and $k = 10$. A retrieved item is counted as correct if it belongs to the same high-level class as the query.

Zero-Shot Classification. Accuracy is measured by assigning each time series to the closest class prompt in the embedding space. Results are reported on both 3 and 9 class setups.

Linear Probing. We assess representation quality by training a linear classifier on frozen time series embeddings. Accuracy is reported for both the 3 and 9 class tasks.

3.3. Results

We report results across three evaluation axes: cross-modal retrieval, zero-shot classification, and linear probing.

1. Cross-modal Retrieval Pair-level recall remains low, but class-level F1 scores indicate meaningful semantic learning. Moirai-l with a non-frozen LLM outperforms all models: Recall@1 is 0.53%, Recall@10 is 2.49%, Recall@100 is 14.3%, and F1@10 reaches 55.5%. Freezing the LLM reduces pair-level recall by an order of magnitude but lowers F1@10 only slightly (from 55.5% to 48.1%), showing that semantic grouping is mostly preserved. Moment mod-

Table 1. Cross-modal retrieval results (%). Recall@k measures exact pair retrieval. Class-level F1@k captures semantic grouping.

Model	Frozen LLM	Recall (pair)			F1-score (class)	
		@1	@10	@100	@1	@10
Moirai-s	no	0.19	0.98	7.1	41.8	51.2
Moirai-s	yes	0.05	0.22	2.14	37.9	46.8
Moirai-l	no	**0.53**	**2.49**	**14.3**	**46.1**	**55.5**
Moirai-l	yes	0.07	0.37	2.67	39.3	48.1
Moment-s	no	0.15	0.95	7.96	32	32.3
Moment-s	yes	0.04	0.23	1.54	29	30.4
Moment-l	no	0.39	2.1	13.35	36.4	37
Moment-l	yes	0.06	0.28	2.49	36.7	37.7

els follow the same trend but perform consistently lower. For example, Moment-l (non-frozen) achieves Recall@1 of 0.39% and Recall@100 of 13.35%, with F1@10 at 37.0%. Freezing its LLM drops Recall@1 to 0.06%, but F1@10 remains nearly the same (37.7%), again suggesting robustness in semantic clustering. As shown in Figure 3, retrieved examples often share the same class and sensor characteristics even when they are not exact matches. Appendix C provides additional qualitative cases, including examples where spelling differences or multiple valid matches lead to underestimated retrieval metrics.

Table 2. Prompt-based zero-shot classification accuracy (%).

Model	3-class	9-class
Moirai-s (non-frozen LLM)	62.9	36.3
Moirai-s (frozen LLM)	**75.7**	41.6
Moirai-l (non-frozen LLM)	52.1	26.3
Moirai-l (frozen LLM)	61.5	21.8
Moment-s (non-frozen LLM)	64.1	32.8
Moment-s (frozen LLM)	63.7	**44.2**
Moment-l (non-frozen LLM)	63.3	24.4
Moment-l (frozen LLM)	64.6	41.9

2. Zero-Shot Classification Freezing the LLM leads to more robust zero-shot generalization, especially for Moirai-s (Table 2). For Moirai-s, keeping the RoBERTa encoder fixed improves 3-class accuracy from 62.9% to 75.7%, and slightly boosts 9-class accuracy from 36.3% to 41.6%. Moment-s (frozen) achieves the highest 9-class zero-shot accuracy at 44.2%, followed closely by Moment-l (frozen) at 41.9%. This suggests that preserving the pretrained semantic priors of the LLM is beneficial, especially for fine-grained prompt-based classification. Moirai-l shows a different pattern: freezing improves 3-class performance (52.1% to 61.5%) but decreases 9-class accuracy (26.3% to 21.8%), indicating potential over-reliance on fixed representations. Overall, fine-tuning the LLM can reduce its ability to generalize to prompts, likely due to operation-specific overfitting.

Table 3. Linear classification accuracy on frozen embeddings (%). MP = multimodal pretraining; Init = original pretrained checkpoint.

Model	Frozen LLM	3-cls		9-cls	
		MP	Init	MP	Init
Moirai-s	no	87.8	**86.9**	**74.2**	**63.9**
Moirai-s	yes	87.3	**86.9**	71.4	**63.9**
Moirai-l	no	**88.5**	79.2	74.1	63.3
Moirai-l	yes	81.5	79.2	71.4	63.3
Moment-s	no	85.5	72.6	66.5	53.9
Moment-s	yes	84.6	72.6	60.7	53.9
Moment-l	no	89.3	65.1	72.2	41.5
Moment-l	yes	88.3	65.1	67.7	41.5

3. Linear Probing Linear probing shows consistent gains from multimodal pretraining (Table 3). Moment-l (non-frozen LLM) reaches 89.3% on the 3-class task and 72.2% on the 9-class task, surpassing its initial checkpoint by 24.2 and 30.7 percentage points. Moirai-s (non-frozen) posts the highest 9-class accuracy overall at 74.2%, a 10.3-point jump over its original weights. While there are notable differences between frozen and non-frozen LLM variants for linear probing, particularly for Moirai-l and Moment-s/l on 9-class tasks, the primary benefit for linear probing accuracy consistently comes from the joint text and time series training itself (MP) rather than solely from LLM fine-tuning. These results confirm that multimodal alignment improves the linear separability of downstream classes in the sensor embedding space.

4. Conclusion

We introduced a multimodal contrastive learning framework that aligns drilling sensor data with textual operations from DDRs. By pairing pretrained LM4TS with a domain-adapted language encoder, our approach improves representation quality beyond what unimodal pretraining offers.

Multimodal training enables capabilities not seen in standard LM4TS models, most notably, zero-shot classification and cross-modal retrieval. Our experiments show that fine-tuning the text encoder improves pair retrieval. The zero-shot classification improves with text encoder being frozen for almost all cases. The linear classification task shows improved performance with the multimodal training weights compared with the initial published weights of LM4TS.

The contrast between pair retrieval and zero-shot results reveals a trade-off between alignment strength and semantic robustness. They also suggest that stronger or carefully staged language models may help reconcile this tension, enabling robust multimodal systems.

Future work includes exploring larger language models, domain-adaptive pretraining strategies, and real-time deployment for drilling advisory systems.

References

Ansari, A. F., Stella, L., Türkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. CHRONOS: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Baldenweg, F., Burger, M., Rätsch, G., and Kuznetsova, R. Multi-modal contrastive learning for online clinical time-series applications. *arXiv preprint arXiv:2403.18316*, 2024.

Buiting, J., Sengupta, S., Gupta, B., Tamaazousti, Y., and Benzine, A. When larger isn't better: Lightweight cnns outperform large time-series models in classification of oil and gas drilling data. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. MOMENT: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Li, J., Liu, C., Cheng, S., Arcucci, R., and Hong, S. Frozen language model helps ecg zero-shot learning. In *Medical Imaging with Deep Learning*, pp. 402–415. PMLR, 2024.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. In *arXiv preprint arXiv:1807.03748*, 2018.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. 2024.

A. Strings used for zero shot classification

- `TRIP`: RIH, POOH
- `DRILL`: DRILL
- `CSG`: CSG, LINER
- `CM`: TUBING, TBG
- `CMT`: CEMENT, CMT, JOB
- `CORE`: CORE, CORING
- `STKP`: STUCK, WORKING STRING
- `DRLOUT`: DRILL OUT
- `REAM`: WASHED DOWN, REAM

B. Training Specifics

Our model's training configuration included the following specifics:

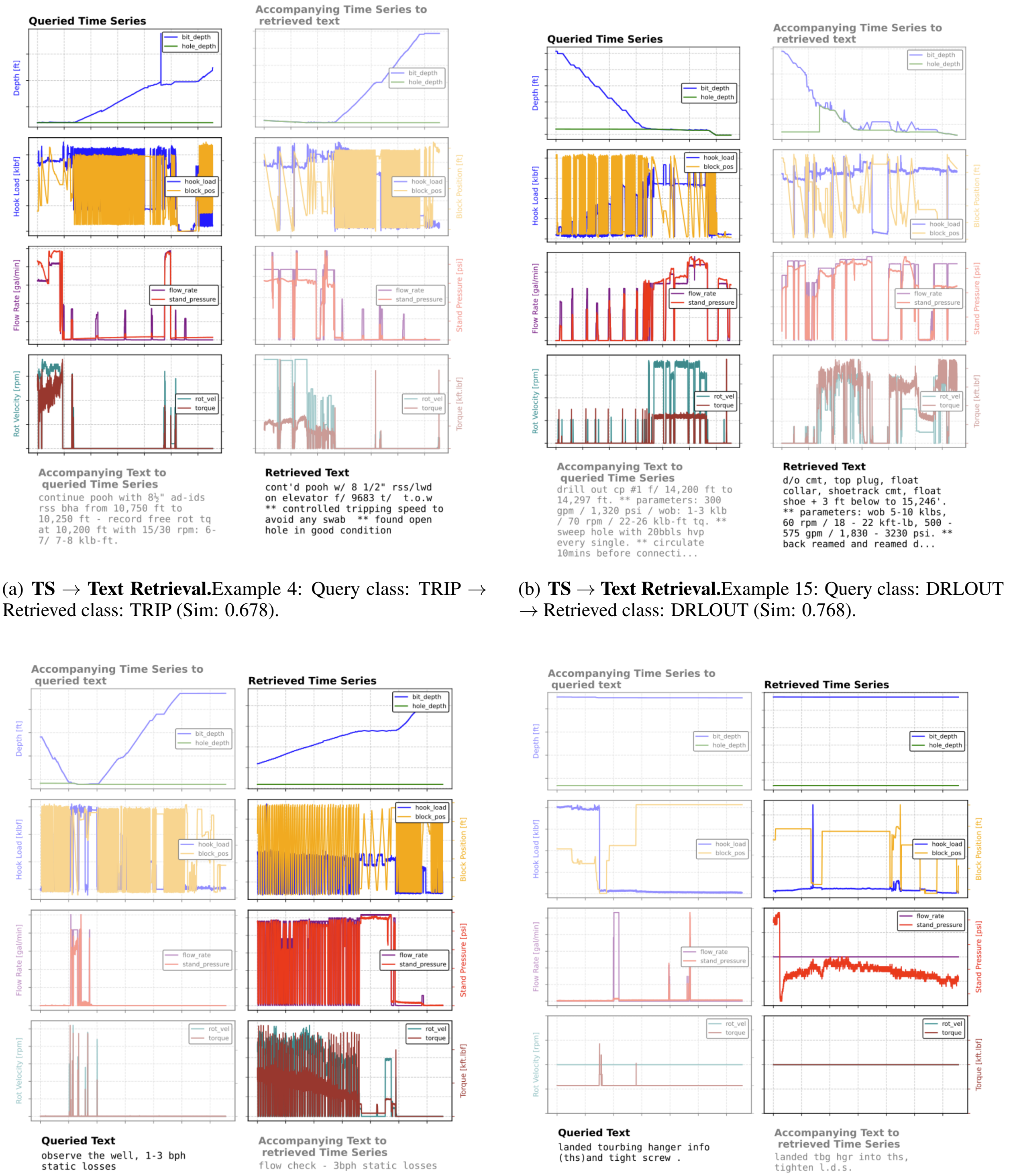
- **Optimizer:** We used the Adam optimizer with a learning rate of 2.5×10^{-5} for multimodal training and 2.5×10^{-4} for the linear probing.
- **Embedding Dimension:** Both the time series and language models project their outputs into a shared embedding space of **256 dimensions**.
- **Pooling Strategies:**
  - The language model's embedding was derived by taking the **last token's representation**.
  - The time series model utilized a **mean pooling** approach to compress its window-length dimension.
- **Early Stopping:** To prevent overfitting on the multimodal model, we monitored the **InfoNCE loss**. Training halted if the loss didn't improve by at least 0.005 for 3 consecutive epochs (mode: 'min').

  For the linear probing model, we monitored the **accuracy**. Training halted if the loss didn't improve by at least 0.002 for 10 consecutive epochs (mode: 'max').
- **Data Preprocessing:** Input sensor data features were normalized using **max-scaling** on each channel.

C. Qualitative Retrieval Examples

Discussion. These examples highlight the practical challenges and strengths of contrastive retrieval in drilling data. In Figure 4(a), the model retrieves a text describing the same class (`TRIP`) and same section size (8.5"). Despite a 10% difference in depth, the match is semantically correct. The model also handles phrasing variations like "cont'd" vs. "continue" and "w/" vs. "with". Figure 4(b) shows a `DRLOUT` retrieval with partial sensor alignment: torque values are close, rotation is reasonably similar (60 vs. 70), but pressure varies more significantly (1320 vs. 1830). In Figure 4(c), the retrieved time series fully matches the query text. The operation (well and flow check) and the observed losses are consistent, showing how one text can align with multiple time series. Figure 4(d) is another such case: the retrieved text is semantically identical, but different spelling causes a false positive under our current evaluation. These examples suggest that the actual semantic retrieval accuracy may exceed what pairwise matching metrics report, due to legitimate ambiguity in real-world annotations.

D. Text Retrieval Example



Accompanying Text to queried Time Series

continue pooh with 8½" ad-ids rss bha from 10,750 ft to 10,250 ft - record free rot tq at 10,200 ft with 15/30 rpm: 6-7/ 7-8 klb-ft.

Retrieved Text

cont'd pooh w/ 8 1/2" rss/lwd on elevator f/ 9683 t/ t.o.w ** controlled tripping speed to avoid any swab ** found open hole in good condition

(a) **TS → Text Retrieval.**Example 4: Query class: TRIP → Retrieved class: TRIP (Sim: 0.678).

Accompanying Text to queried Time Series

drill out cp #1 f/ 14,200 ft to 14,297 ft. ** parameters: 300 gpm / 1,320 psi / wob: 1-3 klb / 70 rpm / 22-26 klb-ft tq. ** sweep hole with 20bbls hvp every single. ** circulate 10mins before connecti...

Retrieved Text

d/o cmt, top plug, float collar, shoetrack cmt, float shoe + 3 ft below to 15,246'. ** parameters: wob 5-10 klbs, 60 rpm / 18 - 22 kft-lb, 500 - 575 gpm / 1,830 - 3230 psi. ** back reamed and reamed d...

(b) **TS → Text Retrieval.**Example 15: Query class: DRLOUT → Retrieved class: DRLOUT (Sim: 0.768).

Queried Text

observe the well, 1-3 bph static losses

Accompanying Text to retrieved Time Series

flow check - 3bph static losses

(c) **Text → TS Retrieval.**Example 39: Query class: TRIP → Retrieved class: TRIP (Sim: 0.744).

Queried Text

landed tourbing hanger info (ths)and tight screw .

Accompanying Text to retrieved Time Series

landed tbg hgr into ths, tighten l.d.s.

(d) **Text → TS Retrieval.**Example 41: Query class: CM → Retrieved class: CM (Sim: 0.725).

Figure 4. **Qualitative Retrieval Examples.** These figures show more examples of cross-modal retrieval results in the drilling domain. Each subfigure displays a query sample (left) and the top-1 retrieved sample (right) along with their similarity score. Subfigures (a) and (b) show time series to text retrieval, while (c) and (d) show text to time series retrieval. Note the specific activity/text snippets and similarity scores provided for each example.