Project Logbook: TrAISformer Reproduction

Project: Reproduction of "TrAISformer: A Transformer Network with Sparse Augmented Data Representation and Cross Entropy Loss for AIS-based Vessel Trajectory Prediction"
Investigator: Polly Chen
Date Range: Dec 23, 2025 – Dec 24, 2025
Objective: Reproduce experimental results from the original paper (Nguyen et al., 2021) using the Danish Maritime Authority (DMA) dataset.

Entry 1: Project Setup & Architecture Review

Date: Dec 23, 2025
Task: Initialization and Codebase Review

Action: Cloned the official repository from GitHub (CIA-Oceanix/TrAISformer).

Code Analysis: Reviewed the model architecture in `trAISformer.py`.

Architecture Attribution: Confirmed the repository implements the "Four-hot" encoding (Lat, Lon, SOG, COG) as described in the paper. I will utilize this existing module rather than implementing it from scratch to ensure exact architectural fidelity.

Dataset Decision:

- **Context:** For this paper, I downloaded the Danish Maritime Authority (DMA) data as the paper used to reproduce the result.
- **Decision:** Switched to the DMA dataset (included in `./data/ct_dma/`) for this training run. This is strictly to validate reproducibility against the paper's reported metrics.

Entry 2: Environment Configuration & Debugging

Date: Dec 23, 2025
Task: Environment Setup on Google Colab (T4 GPU)

Setup: Initialized Google Colab runtime with T4 GPU. Verified via `torch.cuda.is_available()`.

Dependency Issue: The provided `requirements.yml` is for Conda, but Colab uses pip.
Resolution: Manually installed missing dependencies: `!pip install einops tqdm`.

Critical Bug (Python 3.12 Compatibility):

- **Error:** `AttributeError: '_SingleProcessDataLoaderIter' object has no attribute 'next'`
- **Root Cause:** The codebase uses the `.next()` method for iterators, which was removed in Python 3.9+ (Colab currently uses Python 3.10/3.12).
- **Fix:** Applied a global find-and-replace to update the syntax in `trainers.py`.
- **Command:** `sed -i 's/.next()/.__next__()/g' trainers.py`

Entry 3: First Training Attempt (Failure)

Date: Dec 23, 2025
Task: Initial Training Run

Execution: Started training with default configuration.

- **Model Parameters:** ~57.4 Million
- **Training Set:** 9,144 trajectories

Incident: The Colab session timed out and disconnected at Epoch 48.

Consequence: All checkpoints were saved to local Colab storage (`./results/`), which is ephemeral. Data was lost.

Action Item: Must implement persistent storage before retrying.

Entry 4: Implementing Persistence (Google Drive)

Date: Dec 24, 2025
Task: System Integration

Objective: Prevent data loss from session timeouts.

Action: Mounted Google Drive and modified the configuration to save checkpoints directly to Cloud Storage.

Code Modification:

python

```
from google.colab import drive

drive.mount('/content/drive')


# Modified config_trAISformer.py to point to Drive

!sed -i 's|./results|/content/drive/MyDrive/TrAISformer_results|g' config_trAISformer.py

!mkdir -p /content/drive/MyDrive/TrAISformer_results
```

Entry 5: Successful Training Run

Date: Dec 24, 2025
Task: Full Training Reproduction (50 Epochs)

Execution: Reran training with Google Drive persistence. Total duration: ~100 minutes.

Training Dynamics Observation:

Phase	**Epochs**	**Observation**
Convergence	1-5	Rapid loss convergence
Optimal	10	**Best Validation Loss (1.38)** observed

Overfitting	11-50	Training loss continued to decrease (negative values), but Validation loss began to increase (1.38 -> 3.91).

Analysis: The model exhibits clear overfitting after Epoch 10. This aligns with the paper's methodology of using Early Stopping. The model capacity (57M params) is likely large relative to the dataset size.

Outcome: Saved the best model checkpoint from Epoch 10 for testing.

Entry 6: Final Results & Verification

Date: Dec 24, 2025
Task: Evaluation and Unit Conversion

Testing: Evaluated the best model (Epoch 10) on the test set (1,453 trajectories).

Unit Conversion:

- The code outputs metric (Haversine distance) in Kilometers (km)
- The paper reports in Nautical Miles (nmi)
- Conversion Factor: 1 nmi ≈ 1.852 km

Comparative Analysis:

Prediction Horizon	My Result (km)	My Result (nmi)	Paper Table I (nmi)	Status
1 Hour	0.89 km	0.48 nmi	0.48 nmi	✓ Exact Match
2 Hours	1.70 km	0.92 nmi	0.94 nmi	✓ Successful
3 Hours	2.79 km	1.51 nmi	1.64 nmi	✓ Successful

Note: Minor improvements over paper results may be attributed to different random initialization or early stopping point selection.

Conclusion: The reproduction is successful. The 1-hour prediction error matches the State-of-the-Art result reported in the original paper exactly. The 3-hour gap (1.51 vs 1.64) is ~8% better than the paper: Results are within expected variance; minor improvements may be attributed to different random initialization.

Appendix: Complete Reproduction Script

python

```
# Mount Google Drive

from google.colab import drive

drive.mount('/content/drive')


# Clone repository

!git clone https://github.com/CIA-Oceanix/TrAISformer.git

%cd TrAISformer


# Fix Python 3.12 compatibility

!sed -i 's/.next()/.__next__()/g' trainers.py


# Save results to Google Drive

!sed -i 's|./results|/content/drive/MyDrive/TrAISformer_results|g' config_trAISformer.py

!mkdir -p /content/drive/MyDrive/TrAISformer_results


# Run training

!python trAISformer.py
```

Summary of Technical Modifications

Issue	Cause	Solution
`.next()` AttributeError	Python 3.12 removed iterator `.next()`	Replace with `.__next__()`
Results lost on disconnect	Colab local storage is temporary	Save directly to Google Drive
Conda requirements incompatible	`requirements.yml` for conda, not pip	Use Colab pre-installed packages