Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# InlandVessel-2K-Sim: A Reproducible Multi-Source Testbed and Evaluation Protocol for Inland Vessel Trajectory Prediction, with a Spatiotemporal Graph Attention Reference Model

**JunKai Zhong**
Jining Polytechnic
**Yanpeng Li**
Jining Polytechnic
**JianLei Liu**
jianleiliu@qfnu.edu.cn

Qufu Normal University

**Article**

**Keywords:** Inland vessel, trajectory prediction, reproducible benchmark, spatiotemporal graph attention, multi-source data fusion, simulation

**Posted Date:** November 3rd, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11277786/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Abstract

We release **InlandVessel-2K-Sim**, a configurable simulator generating synchronised AIS-radar-vision scenes for two inland channel geometries in the field-recording data layout, with an evaluation protocol anchoring absolute error on constant-velocity extrapolation, averaging three seeds on a fixed split, and scene-level bootstrap and paired tests. The testbed is demonstrated with **ST-GAT-Fusion**, a spatiotemporal graph attention reference model coupling three modality-specific encoders through dynamic-time-warping alignment, with explicit vessel-to-vessel and vessel-to-shore edges. On the released 160-scene split the four fusion-and-graph configurations sit at 19.2–19.5 m ADE — roughly half the 40.20 m of constant-velocity extrapolation — with differences inside seed variability (0.62–0.89 m for fused models, 0.04–0.11 m for AIS-only baselines). We therefore report, as a primary finding, that multi-source fusion yields no resolvable accuracy advantage: a capacity-matched AIS-only Transformer matches the fused model, a fixed-budget data-scale control shows the fusion edge eroding rather than growing with training scenes, and a larger AIS-only Transformer reaches 17.3 m, below all fused configurations. What it delivers is robustness: under AIS outage the fused model loses less than half a metre where the AIS-only baseline loses ten, and with no AIS it still predicts while the baseline cannot. The half-width model sustains 179 vessels/s on CPU.


# 1. Introduction

Inland waterway transport accounts for a substantial and growing share of regional freight movement worldwide. In China, the "Four Vertical, Four Horizontal, Two Network" high-grade waterway system is expected to carry more than 4.5 billion tonnes of cargo annually by 2030, and the Grand Canal and the Yangtze River — the two waterways that frame this work — together host tens of thousands of cargo vessels every day. As traffic density rises, accurate short-term vessel trajectory prediction has become a pressing operational requirement, with direct consequences for collision avoidance, lock and bridge scheduling, search-and-rescue coordination, and the early detection of anomalous behaviour such as drifting or unauthorised berthing.

Conventional inland-vessel trajectory-prediction pipelines rely almost exclusively on the Automatic Identification System (AIS). Recent deep-learning approaches have lifted single-vessel AIS-only forecasting well beyond classical Kalman and particle filters: recurrent encoders such as LSTM, GRU and Bi-LSTM are now routinely used to capture long-range temporal dependencies [1, 2], and more recent studies replace recurrence with Transformer decoders, attention-augmented seq2seq models, or graph structures that treat the vessel as a single node [3, 4, 5]. While these models achieve strong accuracy on vessels that broadcast clean AIS, they share three structural weaknesses: (i) AIS reception is intermittently blocked by bridges, locks and dense urban shorelines, leaving gaps that no temporal model can interpolate reliably; (ii) small craft such as fishing boats, tugs and recreational vessels frequently carry no AIS at all, so an AIS-only model has no signal for them; and (iii) abrupt collision-avoidance manoeuvres are rare in historical data and are typically smoothed away by models trained to minimise long-horizon mean error.

A second line of work addresses some of these gaps by adding radar or visual modalities. Marine radar provides all-weather range and bearing observations independent of vessel cooperation, and modern point-cloud processing supports detection, tracking and re-identification from radar alone [6]; shore-based CCTV along busy inland sections supplies dense appearance and class information that radar cannot [7, 8]. In the autonomous-driving and V2X communities, late, early and intermediate fusion of camera, radar and lidar has matured into production-grade perception stacks, and several studies have explored such fusion for vessel re-identification or port monitoring [9, 10, 11]. Yet these systems are tuned for open sea, port entrances or vehicle highways, where trajectories are short, vessel-to-vessel interactions are sparse and deployment cost is high; none has been adapted to confined inland channels, where the number of nearby vessels is large, their relative motion is strongly correlated, and the time budget for prediction is constrained by the short inter-vessel distance.

This gap motivates the present work. We argue that inland vessel trajectory prediction requires a model that (i) ingests AIS, radar and vision streams jointly rather than choosing among them, (ii) treats vessel-to-vessel and vessel-to-shore interactions as an explicit graph structure rather than as pairwise attention, and (iii) is evaluated on a reproducible multi-modal inland testbed, published together with the field-recording data layout, so that measured data can replace the simulator without code changes.

To this end we build and release **InlandVessel-2K-Sim**, a configurable simulator that emits synchronised AIS–radar–vision scenes for two inland channel geometries in the field-recording layout, and we demonstrate it with **ST-GAT-Fusion**, a spatiotemporal graph attention reference model that couples three modality-specific encoders with a cross-modal temporal alignment module and a spatiotemporal graph attention decoder. We state the contribution of this paper up front, because the empirical outcome shapes how the work should be read: the deliverable is the **testbed and its evaluation protocol**, and applying that protocol reveals that multi-source fusion, in its current form, does not improve accuracy over AIS-only baselines at the scale of the released data. Our contributions are threefold:

1. We construct and release **InlandVessel-2K-Sim**, a synchronised AIS-radar-vision simulator for two inland channel geometries that emits data in the field-recording layout, together with the complete training, ablation and robustness pipeline, to support reproducible research on multi-modal inland prediction.
2. We define and release an **evaluation protocol** for such testbeds: a parameter-free constant-velocity anchor that fixes the absolute error scale, a fixed three-way split, three-seed averaging with seed dispersion, and scene-level bootstrap confidence intervals and paired Wilcoxon tests. Applying this protocol yields the **central empirical finding of this study, which we report as a result rather than a caveat**: on the released split, multi-source fusion produces no resolvable accuracy advantage over AIS-only baselines, whereas the degradation robustness that the multi-source design does deliver under AIS loss is large and reproducible.
3. As a reference implementation we release **ST-GAT-Fusion**, which aligns and jointly reasons over AIS, radar and vision streams through masked cross-modal attention, and whose spatiotemporal graph

attention layer models vessel-to-vessel and vessel-to-shore coupling explicitly; we isolate that layer's contribution against edge-free and single-modality variants.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 reports the benchmark, the evaluation protocol and all quantitative results, including the controls, ablations, robustness and generalisation analyses. Section 4 discusses conclusions, limitations and future work. Section 5 (Methods) formalises the prediction problem, details the ST-GAT-Fusion architecture, and specifies the evaluation metrics and the baselines.

## 2. Related Work

Three research streams are relevant to the inland-vessel prediction problem: single-vessel forecasting on AIS, multi-modal maritime perception, and graph-based spatiotemporal modelling.

# 2.1 AIS-based Trajectory Prediction

Recurrent encoders, particularly LSTM and GRU, became the dominant baselines for AIS forecasting after 2018 thanks to their ability to model long temporal dependencies in vessel motion [1, 2], with attention pooling [12] and social pooling across interacting vessels [13] adding interaction context. Transformer architectures [3] and multi-modal heterogeneous forecasters such as Trajectron++ [4] now match or surpass RNN baselines on clean AIS traces; sequence-to-sequence models [14], generative transformers over sparse AIS representations [15], anomaly detection on learned probabilistic track representations [16] and convolutional predictors with auxiliary tasks [17] push in the same direction, and two recent surveys map the resulting literature [18, 19]. These models share one structural assumption: the vessel under prediction is the only source of information. They therefore offer no signal for AIS dropouts near locks and bridges, for the many small craft that carry no AIS transponder, or for abrupt collision-avoidance manoeuvres that historical long-horizon training tends to smooth away.

# 2.2 Multi-Modal Vessel Perception and Fusion

A second body of work augments or replaces AIS with radar and optical sensors. Operational vessel traffic services have long combined radar tracks with AIS labels, but their fusion logic is hand-crafted and installation-specific; modern pipelines instead re-cast radar tracking with PointNet++- and CenterPoint-style backbones [6], while YOLO-family detectors with re-identification heads reach high maritime re-identification accuracy [7, 8]. Radar–camera fusion has appeared in bridge and lock monitoring [9]; on inland waterways specifically, AIS–LiDAR association with adaptive filtering [20], attention-based predictors for bridge-crossing manoeuvres [21], improved Transformer encoders for constrained reaches [22], vision–LiDAR stacks for port-channel navigation [23], dedicated small-craft detectors [24] and environment-augmented AIS predictors [25] have all been demonstrated. In autonomous driving, BEVFusion [10] and TransFusion [11] standardised early, late and intermediate fusion. All of these target open sea, port entrances or vehicle highways where trajectories are short and interactions sparse; the

confined-channel geometry — high vessel density, strongly correlated relative motion, short prediction horizons — has not been systematically studied.

## 2.3 Graph-Based Spatiotemporal Modelling

A third body of work models transportation entities as nodes of a dynamic graph: ST-GCN introduced spatiotemporal convolutions on road-segment graphs [5], Graph WaveNet and GMAN added adaptive adjacency and attention-based spatial aggregation [26, 27], and the stacked-transformer mmTransformer [28] is representative of multi-agent motion forecasting in V2X settings. Graph attention [29] and graph convolution [30] supply the core operators; spatiotemporal graph models have been applied to interacting pedestrians [31, 32] and to ship trajectory prediction [33], with hierarchical vector transformers [34, 35] and simple attention-based forecasters [36] as strong motion-forecasting baselines. Dynamic time warping remains the standard tool for aligning irregularly sampled sequences [37], and graph representations have been combined with uncertainty estimation for maritime anomaly detection [38]. No existing graph-based predictor, however, has been adapted to the inland regime, where AIS samples arrive at 2–10 s rather than 30–60 s intervals, radar points are sparse but dense per scan, and vessel interactions are tightly coupled by narrow-channel geometry rather than by road topology.

# 3. Results

## 3.1 Simulated benchmark: InlandVessel-2K-Sim

We release **InlandVessel-2K-Sim**, a simulator that generates synchronised AIS-radar-vision scenes in exactly the data layout used by the field recordings (stgat_fusion/data/scene.py). It models two channel geometries chosen to bracket the inland range:

Two channel geometries bracket the inland range: a GC-JN style reach (channel half-width 90 m, bridge spacing 1.5–2.5 km, curve amplitude 60 m, 6% fog-induced misses) and a wider YZ-NJ style reach (150 m half-width, 2.5-4.0 km bridge spacing, 140 m curve amplitude, 10% fog-induced misses).

Table 1
**InlandVessel-2K-Sim statistics.** Scenes are generated on a 1 s grid with 8 observation steps and 12 prediction steps. Availability is the fraction of per-vessel time steps at which a source produced an observation.

| Subset | # Scenes | # Vessel trajectories | AIS avail. | Radar avail. | Vision avail. |
|---|---|---|---|---|---|
| GC-JN style | 80 | 711 | 68.5% | 83.7% | 75.5% |
| YZ-NJ style | 80 | 724 | 70.8% | 87.6% | 75.8% |
| **Total** | **160** | **1,435** | **69.6%** | **85.7%** | **75.7%** |

At the level of individual vessel steps, all three sources are simultaneously available for 47.6% of steps and at least one source is available for 96.8%; 2.8% of steps have AIS only (Fig. 1). This is the regime the modality mask of Section 5.4 is designed for.

The impairment model is deliberately explicit so that each degradation can be switched off or re-parameterised for a different waterway. AIS reports arrive at 2–10 s intervals, are resampled to 1 Hz, carry AR(1) position noise (σ = 4 m) and a 2% outlier rate, and drop out per vessel with availability drawn from 0.75–1.0 plus occasional 15–35 s outages. Bridges impose a 55% chance of a 10–30 s simultaneous AIS/radar/vision shadow on crossing vessels inside a ± 150 m zone. Radar is sampled at 0.5 Hz with point counts scaled by vessel length (up to 64 points), plus clutter and a 10% dropout rate; vision is sampled at 0.5 Hz with AR(1)-perturbed identity embeddings, a 12% dropout rate and a channel-dependent fog miss rate. Vessel kinematics use an Ornstein-Uhlenbeck speed process, a limited turn rate, moored vessels and bidirectional lanes, all in a local metric frame.

**Scope and limitation.** The released split contains 160 scenes (112 train, 24 validation, 24 test), which at batch size 16 yields seven mini-batches per epoch, so every model in this section is lightly fitted; the seed coverage of each study is stated where it appears, and the results characterise the pipeline on a controlled testbed rather than estimating field accuracy.

# 3.2 Main Results

Table 2

**Quantitative comparison on InlandVessel-2K-Sim.** Mean ± s.d. over three seeds (0, 1, 2); the released split (112/24/24 scenes) is held fixed across seeds, so the dispersion shown is training variability on identical data. Test metrics come from the validation-selected checkpoint of each run; throughput is a single seed-0 inference pass on CPU. The constant-velocity row is parameter-free, has no seed, and is evaluated with the same metric implementation on the same 24 test scenes. The two AIS-only Transformer rows are analysed separately in the capacity ladder at the end of this section.

| Method | ADE (m) ↓ | FDE (m) ↓ | MR@2m (%) ↓ | HE (°) ↓ | Params | Throughput (vessels/s) ↑ |
|---|---|---|---|---|---|---|
| Constant velocity (last two AIS fixes) | 40.20 | 70.48 | 100.0 | 77.5 | 0 | — |
| Bi-LSTM (AIS only, h = 64) | 19.49 ± 0.04 | 31.90 ± 0.10 | 99.8 ± 0.3 | 68.21 ± 0.06 | 0.17 M | 2,949 |
| Social-STGCNN (AIS only) | 19.35 ± 0.11 | 31.71 ± 0.20 | 100.0 ± 0.0 | 68.66 ± 0.38 | 0.59 M | 1,553 |
| ST-GAT-Fusion (full, d = 128) | 19.33 ± 0.89 | 32.44 ± 1.90 | 99.1 ± 0.8 | 68.89 ± 1.52 | 1.47 M | 105 |
| ST-GAT-Fusion-Tiny (d = 64) | 19.17 ± 0.62 | 32.01 ± 1.39 | 99.6 ± 0.6 | 67.87 ± 0.66 | 0.40 M | 179 |
| Transformer (AIS only, d = 128) | 18.81 ± 0.31 | 30.05 ± 0.74 | 99.3 ± 0.6 | 74.07 ± 2.13 | 0.29 M | 1,908 |
| Transformer (AIS only, d = 384) | 17.33 ± 0.34 | 27.02 ± 0.49 | 99.6 ± 0.3 | 74.07 ± 1.57 | 4.79 M | 944 |

The constant-velocity baseline anchors the absolute scale: naive extrapolation reaches 40.20 m ADE and 70.48 m FDE, so every learned configuration roughly halves the ADE and more than halves the FDE of the

naive predictor — the one unambiguous separation in the table. Among the four fusion-and-graph configurations the ADE spread is 0.32 m (19.17–19.49 m), far smaller than the fused models' seed-to-seed standard deviations (0.62 m Tiny, 0.89 m full-width) and much larger than the AIS-only recurrent baselines' (0.04–0.11 m), so we claim no significant accuracy advantage for multi-source fusion, consistent with the scene-level paired test below (p = 0.060). The full-width model's deficit relative to Tiny persists across seeds, consistent with the data-limited regime of Section 3.1, and is why the ablations of Section 3.3 use Tiny as their subject.

The AIS-only baselines are markedly more stable across seeds (s.d. 0.04–0.11 m) than the fused models (0.62–0.89 m), indicating that the fusion modules are under-constrained at this data scale; per-seed metrics are written by tools/run_multiseed.py to results/multiseed_summary.json in the released package. On the seed-0 predictions the per-scene outcome favours the fused model in 17 of the 24 test scenes (Fig. 4a), but that paired test does not reach significance and the direction is not stable across seeds.

**Scene-level dispersion and paired tests.** Table 2 pools vessels across scenes; to expose scene-level sampling uncertainty we additionally compute, on the seed-0 predictions, per-scene mean ADE over the 24 test scenes (each scene weighted equally), bootstrap 95% confidence intervals over scenes (10,000 resamples) and Wilcoxon signed-rank tests on paired scene errors. Tiny averages 17.64 ± 4.68 m per scene against 18.52 ± 4.65 m for the Bi-LSTM baseline, winning 17 of 24 scenes with p = 0.060 — a consistent direction that does not reach significance. Against the constant-velocity baseline the separation is unambiguous: Tiny wins 24 of 24 scenes (mean paired difference − 23.8 m ADE, $p < 10^{-6}$), as does Bi-LSTM (− 22.9 m). All dispersion statistics are produced by tools/eval_cv_stats.py.

Throughput spans more than an order of magnitude (2,949 vessels/s for Bi-LSTM against 105 for the full-width model), but even the slowest configuration is far above the rate at which vessels enter a scene, so inference cost is not binding; within the fusion family Tiny remains the recommended default, and the most accurate entry overall, the $d = 384$ AIS-only Transformer, is also nine times cheaper at inference than the full-width fused model.

**Data-scale control.** The comparison above leaves open the standard objection that the fusion modules simply need more data than the 112 training scenes provide. We therefore repeated the Tiny-versus-Bi-LSTM comparison at 25%, 50% and 100% of the released training scenes, holding validation and test scenes fixed at the released 24/24 and the optimisation budget fixed at 300 gradient updates per run, with epochs and every epoch-based schedule parameter rescaled so that each point receives the same number of updates; the absolute numbers of this control are therefore not comparable with Table 2.

Table 3

**Data-scale control: does the fusion advantage emerge with more data?** Test ADE (m), mean ± s.d. over two seeds (1, 2); validation and test scenes are the released 24/24 at every point, and every run receives the same 300-update budget.

| Training scenes | ST-GAT-Fusion-Tiny ADE (m) | Bi-LSTM (AIS only) ADE (m) | ΔADE (Tiny − Bi-LSTM, m) |
|---|---|---|---|
| 25% (28 scenes) | 19.25 ± 0.34 | 19.96 ± 0.02 | −0.71 |
| 50% (56 scenes) | 19.35 ± 0.14 | 19.75 ± 0.02 | −0.40 |
| 100% (112 scenes) | 19.52 ± 0.46 | 19.47 ± 0.03 | + 0.05 |

The control answers the objection in the unexpected direction (Fig. 2). The AIS-only baseline improves monotonically with data (19.96 m to 19.47 m) whereas the fused model does not (19.25 m to 19.52 m), so the paired gap moves from − 0.71 m through − 0.40 m to + 0.05 m and changes sign at the full split. No point produces a fusion advantage that exceeds seed noise: the largest low-data margin, 0.71 m at 25%, is of the same order as the 0.34 m seed spread at that point. The trend direction is the informative part, because it is the opposite of the data-starvation hypothesis: whatever small edge the fused model shows at reduced scale appears while the baseline is still under-fitted and erodes as the baseline catches up. Enlarging the split alone will therefore not make the fusion advantage resolvable; what would is a regime in which radar and vision carry information the AIS stream does not already contain — exactly the condition the simulator's upstream association assumes away (Section 4.2, item vii).

**Observation-noise control and the error floor.** A second objection is that observation noise may pin every model at a noise-imposed floor. We therefore generated a second testbed in which every scene keeps bit-identical vessel poses, occlusion schedule, radar and vision channels and split membership, while the AIS observation noise (AR(1), σ = 4 m per axis, 2% outliers) is switched off — a strict single-variable comparison. Supplementary Table S5 reports both arms, each trained from scratch under the same 300-update budget with two seeds.

Three readings follow (Supplementary Table S5). First, the learned models already absorb most of the observation noise: removing it is worth 0.9–1.3 m to them but 6.8 m (17%) to the constant-velocity anchor, and feeding the noise-free channel to the noisy-trained checkpoints recovers essentially the same gain as retraining. Second, the learned models still sit at 17.6–18.0 m once that nuisance is removed, so the near-tie of Table 2 is not imposed by observation noise; whether the residual is intrinsic motion stochasticity or a limitation of the models is settled by the capacity ladder below, and it points to the models. Third, the fusion difference never survives the seed spread in either arm, so the negative finding is robust to removing the noisiest input channel; incidentally, the fused model's seed spread contracts from 0.62 m to 0.11 m on noise-free AIS, so its instability is driven by the AIS channel rather than by the fusion modules. The probe testbed is generated afresh, and its constant-velocity anchor (39.51 m against 40.20 m on the released testbed) confirms comparable difficulty.

**AIS-only Transformer baseline and the capacity question.** Table 2 also reports a pure-AIS Transformer encoder — the same 5-D kinematic window, no radar, no vision, no graph — at four capacities from 78 K to 4.79 M parameters (Table 4). Its encoder is architecturally unrelated to the reference model's fusion stack, making it a clean check on whether fusion is worth anything at comparable model size, and whether the ~ 19 m band the four released configurations share is a property of the task or of the models.

Table 4

**AIS-only Transformer capacity ladder.** Mean ± s.d. over three seeds (0, 1, 2) on the fixed released split; throughput is a single seed-0 inference pass on CPU. Every row uses the same schedule as Table 2 (100 epochs, Adam at $1 \times 10^{-3}$, learning-rate steps at epochs 40 and 70, teacher annealing over 60 epochs), so the ladder and the fused models are protocol-matched.

| Configuration | Params | ADE (m) ↓ | FDE (m) ↓ | Throughput (vessels/s) ↑ |
|---|---|---|---|---|
| $d_{\text{model}} = 64$, 2 layers, 4 heads | 78,744 (0.08 M) | 18.91 ± 0.26 | 30.53 ± 0.92 | 1,929 |
| $d_{\text{model}} = 128$, 2 layers, 4 heads | 285,336 (0.29 M) | 18.81 ± 0.31 | 30.05 ± 0.74 | 1,908 |
| $d_{\text{model}} = 256$, 4 layers, 8 heads | 2,145,944 (2.15 M) | 17.31 ± 1.80 | 27.09 ± 4.11 | 1,350 |
| $d_{\text{model}} = 384$, 4 layers, 8 heads | 4,790,168 (4.79 M) | 17.33 ± 0.34 | 27.02 ± 0.49 | 944 |

**At matched capacity the negative result is unchanged.** The 0.29 M row is the one to set against the fused models: at the scene level it is indistinguishable from both (Wilcoxon over 24 scenes: ΔADE = + 0.21 m against Tiny, p = 0.60; −0.26 m against the full-width model, p = 0.47), and its three-seed spread (0.31 m) is smaller than Tiny's. A plain attention encoder over AIS alone therefore buys nothing over multi-source fusion at comparable capacity — the same conclusion Table 2 reaches, obtained with a model that shares neither the modality encoders nor the graph.

**Capacity, not data or sensors, sets the 19 m band.** The ladder is flat from 78 K to 285 K (18.91 → 18.81 m), steps down 1.5 m by 2.15 M, and is flat again to 4.79 M (Fig. 3). The 4.79 M model reaches 17.33 ± 0.34 m, the lowest ADE in this study and 1.84–2.16 m below every entry of Table 2: all three of its seeds fall below all three seeds of every fused configuration and of both AIS-only recurrent baselines, significantly so under the seed-level Welch test ($p \le 0.047$) and at the scene level against three of the four ($p \le 0.012$, not against Tiny, $p = 0.21$ — the more conservative test at $n = 24$). This settles what the noise control could not: the residual shared by the four released configurations is not a task-intrinsic floor, because a model of the same input reaches two metres below it on capacity alone.

# 3.3 Ablation Study

We ablate on the Tiny configuration, on which the released matrix was run, and report three complementary views, each trained from scratch with the identical 100-epoch schedule on seed 0, with test metrics from the validation-selected checkpoint. The ablations are single-seed and should be read against the 0.62 m seed-to-seed spread of the fused models (Table 2): the effects highlighted below (up to 2.5 m ADE) are several times that spread and therefore survive seed variation, whereas differences below ~ 0.6 m between neighbouring variants are not resolvable and are reported as neutral rather than as gains.

The three views are reported in Supplementary Table S2 (parts A, B and C). The graph edges are the only module with an unambiguous positive contribution: enabling the spatial, temporal and shore edges recovers 1.07 m (A2 to A3), several times the 0.62 m seed spread of the fused models. On these seed-0 runs A3 also lands 1.06 m below the A0 baseline, although the seed-averaged gap between the fused models and the AIS-only baselines is only 0.2–0.3 m (Table 2), so that margin should not be read as an architecture effect. The two encoder additions are neutral within noise: adding the vision branch on top of A0 costs 0.22 m (A1), and restoring the radar input returns the model to 19.54 m (A2), within 0.01 m of A0 — both inside the seed spread and therefore not resolvable.

**Per-modality ablation.** Removing one modality at a time from A3, by masking its input rather than deleting its encoder so that the parameter count is unchanged, gives a different picture from the modular ladder (Supplementary Table S2, part B): radar is the most load-bearing input (removing it costs 2.53 m) and vision the second (1.86 m), while removing AIS *improves* ADE by 0.55 m. The most likely explanation is that the AIS stream is the noisiest input: it carries AR(1) position noise and a 2% outlier rate, and a single high-value outlier propagates directly into a decoder that predicts increments relative to the last observed state, whereas the radar and vision channels are comparatively clean once the simulator has associated them to vessels. The ablation therefore measures the reliability of the simulator's sensor models as much as the architecture — exactly the kind of effect that field recordings would re-order.

**Alignment ablation.** The released pipeline aligns radar and vision to the AIS grid offline during preprocessing, so alignment cannot be ablated with a configuration flag; instead we train the same model on a deliberately de-synchronised copy of the dataset in which the radar stream is delayed by 1–3 steps and the vision stream by 1–2 steps (Supplementary Table S2, part C). The aligned model reaches 18.47 m against 18.70 m for the de-synchronised one, so alignment is worth 0.23 m (1.2%), the smallest of the three contributions.

# 3.4 Robustness to Modality Degradation

The modality mask of Section 5.4 is exercised during normal training, but the resulting robustness has so far been measured only in aggregate. We therefore degrade the AIS stream of each test vessel at a rate $\rho \in \{0, 20, 40, 60, 80, 100\}\%$ before inference, while radar and vision remain intact, in two ways: (i) *random dropout*, suppressing individual AIS reports uniformly at random, and (ii) *burst occlusion*,

removing contiguous blocks of 5–20 s to mimic bridge and lock shadowing. Suppressed entries are flagged as absent in the modality mask rather than imputed; complete results for both schedules are reported in Supplementary Table S1.

Three findings follow. First, the fused model degrades gracefully: over $\rho = 0 \rightarrow 100\backslash\%$ its ADE moves from 18.45 m to 18.90 m under random dropout and to 18.89 m under burst occlusion, whereas the AIS-only baseline has already lost 10.5 m by $\rho = 80\backslash\%$ and the radar/vision-masked fusion variant degrades by 18.8 m; under burst occlusion, the schedule that matches bridge shadowing, the fused model at $\rho = 60\backslash\%$ (18.37 m) is still more accurate than the AIS-only baseline at $\rho = 0\backslash\%$ (19.54 m). Second, at $\rho = 100\backslash\%$, where a vessel carries no AIS at all, the fused model still produces bounded predictions from radar and vision alone, whereas the AIS-only baseline has no input and cannot be evaluated — precisely the small-craft scenario that motivates the multi-source design in Section 1, and the clearest positive result in this paper. Third, masked attention does not drive the softmax into a degenerate state: unavailable key slots are filled with $-\infty$ before normalisation (Section 5.4), so the failure mode is a gradual loss of precision rather than a numerical collapse, and heading error stays within 66–69 degrees across the sweep. The sweep uses the seed-0 Tiny checkpoint; the qualitative pattern is architectural rather than seed-dependent, but the exact numbers are single-seed.

## 3.5 Generalization Analysis

We evaluate cross-channel transfer by training on one channel style and testing on the other, in both directions (Supplementary Table S3). This is the most demanding generalisation setting available on this testbed.

The direction of transfer dominates the choice of model. Everything trained on the narrow, bridge-dense GC-JN style and tested on the wide YZ-NJ style lands within a 0.9 m band (22.67 / 22.80 / 21.92 m ADE for Bi-LSTM / Social-STGCNN / Tiny) with Tiny marginally ahead; the reverse transfer separates the models by 1.39 m, but with Tiny *behind* both AIS-only baselines (18.21 m against 16.82 m and 16.92 m). The gap between the two directions is 3.7 m for Tiny, larger than any between-model difference within a direction.

We therefore do not claim the cross-channel robustness that a channel-agnostic graph was designed to provide. Two things plausibly explain the reversal: the target channel in the second block is the narrower and more bridge-dense one, so a model that has learned wide-channel interaction distances faces a genuine domain shift; and the source split supplies only 80 scenes, so the fused model, which must learn three encoders plus the graph from seven mini-batches per epoch, has more to lose from that shift than a single-modality baseline does. Distinguishing the two explanations requires more data than the released split provides. All transfer numbers are single-seed; the 3.7 m direction gap and the 1.39 m between-model separation exceed the ~ 0.6 m seed spread, so the reversal is unlikely to be a seed artefact.

## 3.6 Qualitative Behaviour

## 3.7 Lightweight Variant and Complexity Analysis

The full-width model is 8.7x larger than the AIS-only baseline and 3.6x larger than Tiny, yet it is *less* accurate than Tiny (Table 2); growing the AIS-only Transformer from 0.29 M to 2.15 M parameters buys 1.5 m of ADE, so capacity is not intrinsically useless — it simply did not help the fused configurations within the range they were released at. Because Tiny shares the fusion, masking and graph-topology modules of the full model and differs only in hidden width ( $d = 64$), attention-head count and encoder capacity, the released ablation matrix was run entirely on Tiny: the modular and per-modality ablations of Section 3.3 and both cross-channel directions of Section 3.5 all use it, and the complete per-configuration results, including two deviations from the in-domain ordering, are given in Supplementary Tables S2 and S3.

**Statistical robustness.** Tables 2 and 4 report mean ± s.d. over three seeds on a fixed split, resolving training variability but not split variability; Section 3.2 additionally exposes scene-level uncertainty through standard deviations, bootstrap confidence intervals and paired Wilcoxon tests. The two accounts agree within the fusion block, and the capacity ladder breaks the tie from outside: the 4.79 M Transformer separates from three of the four at the scene level and from all four under the seed-level Welch test, while its 0.29 M sibling is indistinguishable from both fused models. Replicating the ablations, the outage study and the upper end of the ladder with a capacity-matched fused model remains the most valuable extension of this work.

# 4. Discussion

# 4.1 Conclusion

This work addresses a practical yet under-studied problem: forecasting the trajectories of inland vessels when AIS alone is insufficient, and measuring progress on a testbed that other groups can reuse. Our contribution is threefold.

First, we construct and release **InlandVessel-2K-Sim**, a multi-source simulator for two inland channel geometries that emits data in the field-recording layout, together with the complete training, ablation and robustness pipeline, which we hope will serve as a community testbed for future inland-vessel research.

Second, we define and release an **evaluation protocol** for such testbeds — a parameter-free constant-velocity anchor, a fixed split, three-seed averaging with seed dispersion, and scene-level bootstrap and paired tests — whose application produces the central empirical finding of this study: at the scale of the released split, multi-source fusion yields no resolvable accuracy advantage over the AIS-only baselines. We report this negative result as a finding rather than a caveat: it is repeatable across seeds, it is measured against an absolute anchor, and it is the kind of evidence a reusable testbed exists to produce. Three controls strengthen it: the data-scale control shows the fused model's small low-data edge eroding — not growing — with more training data (Section 3.2, Table 3); the observation-noise control

shows the shared residual is not sensor noise (Section 3.2, Table S5); and the capacity ladder shows that the same input driven to 4.79 M parameters reaches 17.33 ± 0.34 m, two metres below every fused configuration, so the band the released models occupy is a capacity limit rather than a property of the task (Section 3.2, Table 4).

Third, as a reference implementation we release **ST-GAT-Fusion**, whose spatiotemporal graph attention layer makes vessel-to-vessel and vessel-to-shore couplings explicit and is the only module whose contribution survives the ablation of Section 3.3.

On the released benchmark every learned configuration sits at 19.2–19.5 m ADE — roughly half the 40.20 m of constant-velocity extrapolation, an unambiguous separation (24 of 24 scenes, $p < 10^{-6}$) — while the differences among the four released configurations are not resolvable. What the study establishes is the testbed and the protocol themselves, together with five reproducible results: the absolute scale of the problem; the degradation behaviour under AIS loss, where the fused model loses less than half a metre while the AIS-only baseline loses ten and becomes unevaluable when AIS is absent; the data-scale control (low-data edge eroding, not growing); the observation-noise control (residual not imposed by AIS noise); and the capacity ladder (a matched-size pure-AIS Transformer matches the fused model, a larger one reaches two metres below it).

# 4.2 Limitations

The study carries eight limitations. (i) Every number is computed on simulated data; the simulator reproduces the *shape* of inland impairments but not their measured statistics, so absolute accuracies must not be read as field performance. (ii) The released split is small (160 scenes, seven mini-batches per epoch), so all models are lightly fitted; main results average three seeds on a fixed split and Section 3.2 adds scene-level dispersion and paired tests, but the split itself is not replicated and the ablations, outage study and transfer study remain single-seed, so no accuracy advantage for fusion is claimed at this scale. (iii) The heading-error column is computed from per-step displacement directions of noisy simulated tracks rather than reported courses, inflating absolute HE for every method; the constant-velocity row (77.5 degrees) is the reference point separating this protocol artefact from model behaviour. (iv) Only three trained baseline families are included, and the strongest is a single architecture swept over capacity rather than a survey of the state of the art. (v) The two channel geometries do not exhaust the inland range; mountainous rapids, tidal estuaries, passive radar targets and small unpowered craft are not modelled. (vi) The default predictor is deterministic and returns no calibrated confidence; quantifying predictive uncertainty (Section 5.7) is necessary before autonomous alarm triggering. (vii) Radar and vision are associated to vessels upstream using AIS-reported positions (Sections 5.3.2 and 5.3.3), so the outage study measures compensation for a missing AIS *channel*, not label-free association, which the simulator assumes away. (viii) The comparison of Table 2 is not capacity-matched at the top of the range: whether a fused model scaled to roughly 5 M parameters under a size-specific schedule would close the 1.84 m gap to the best AIS-only Transformer is untested, so the negative fusion result should be read as a statement about the released configurations and about capacity-matched pairs, not about multi-source fusion at every scale.

## 4.3 Future Work

Five directions follow. First, coupling the predictor with a large-language-model interface could translate trajectory deviations into natural-language explanations for VTS operators. Second, distilling ST-GAT-Fusion onto ARM-based edge devices would enable real-time on-lock analytics. Third, integrating the predictor with AIS gap-filling and anomaly-detection services could convert the prototype into a production-grade VTS module. Fourth, the probabilistic $K$-candidate decoder of Section 5.7 would let operators threshold warnings by risk and mitigate the deterministic over-confidence of limitation (vi). Fifth, and most directly implied by limitation (viii), the fused configurations should be re-run at a capacity matched to the strongest AIS-only Transformer with a size-specific schedule, to test whether the multi-source machinery recovers the 1.84 m that separates it from the best single-source model.

# 5. Methods

## 5.1 Problem Formulation

Let $\mathcal{V} = \{v_1, \dots, v_N\}$ denote the set of vessels visible in the inland waterway scene during an observation window, and let $\mathcal{S} = \{s_1, \dots, s_M\}$ denote the set of $M$ channel-bank samples that describe the local waterway geometry; these supply the shore context used in Section 5.5. For each vessel $v_i$, the multi-source observation over $T$ discrete time bins is

$$\mathcal{O}_i = \{o_{i,t}\}_{t=1}^{T}, \;\; o_{i,t} \in \mathbb{R}^{d_{\text{obs}}}$$

where $d_{\text{obs}}$ depends on the available modalities. AIS contributes 5 features (local east and north offsets, speed over ground, and the cosine and sine of course over ground); radar contributes 3 features per point (local east and north offsets and radial Doppler velocity) per point cloud of $K$ points; vision contributes a 128-D appearance embedding per detection. The full shore-side observation is $\mathcal{O}^{\mathcal{S}} = (\mathcal{O}^{\mathcal{S}}_1, \dots, \mathcal{O}^{\mathcal{S}}_M)$.

The prediction target for each vessel is a future trajectory of $H$ steps, expressed in the same scene-local metric frame:

$$\mathcal{Y}_i = [(\, x_i^{T+h},\, y_i^{T+h})]_{h=1}^{H}$$

The full problem is to predict $\mathcal{Y} = (\mathcal{Y}_1, \dots, \mathcal{Y}_N)$ from the joint observation $\mathcal{O} = (\mathcal{O}_1, \dots, \mathcal{O}_N, \mathcal{O}^{\mathcal{S}})$.

**Notation.** $N$, $M$: number of vessels and channel-bank context samples; $T$, $H$: observation horizon (8 steps) and prediction horizon (12 steps); $\Delta t$: sampling interval of the common grid (1 s); $d_{\text{obs}}$: per-step observation dimension (modality-dependent); $d$: shared hidden dimension (128 full-width, 64 Tiny); $K$: radar points retained per vessel and step ($\leq 64$); $\mathcal{G}_t = (\mathcal{V}_t, \mathscr{E}_t)$: spatiotemporal graph at time $t$; $m_i \in \{0,1\}^3$: modality availability mask.

**Working assumptions.** (i) *Spatial locality.* Vessel interactions beyond a critical distance $d_c = 150\,\mathrm{m}$ are negligible for the prediction horizon; this value is set by the released configuration rather than estimated from a measured encounter distribution, and Section 5.5 shows the model's weak sensitivity to it. (ii) *Asynchronous and incomplete modalities.* Within a sliding window each modality may be missing for individual vessels and steps; we model this explicitly with a binary modality mask $m_i$ rather than imputing or discarding incomplete samples. (iii) *Fixed horizon.* $H = 12$ steps ( $12\,\mathrm{s}$ at a $1\,\mathrm{s}$ sampling interval) reflects the decision budget for collision-avoidance and lock-scheduling advisories on the modelled reaches.

# 5.2 Overall Framework

ST-GAT-Fusion is organised as a four-stage pipeline (Supplementary Fig. S1). **Stage A** consists of three modality-specific encoders that map each vessel's AIS sequence, radar point cloud, and vision detections into a shared $d$-dimensional embedding space. **Stage B** performs cross-modal temporal alignment through dynamic time warping (DTW), applied offline at preprocessing time, followed by a 3-way masked cross-modal attention that handles asynchronous and missing modalities. **Stage C** constructs a spatiotemporal graph $\mathcal{G}_t = (\mathcal{V}_t, \mathscr{E}_t)$ in which nodes are vessels plus channel-bank context samples and edges encode spatial proximity, identity persistence, and shore coupling. **Stage D** stacks $L = 2$ spatiotemporal graph attention layers whose outputs feed a Transformer decoder that produces multi-step trajectory predictions. The framework is trained end-to-end with the joint loss described in Section 5.8.

The four stages are detailed in Sections 5.3–5.7. Supplementary Fig. S1 sketches the pipeline; Supplementary Fig. S2 zooms into the encoder trio; Supplementary Fig. S3 illustrates cross-modal alignment; Supplementary Fig. S4 shows the graph structure.

# 5.3 Modality-Specific Feature Extraction

# 5.3.1 AIS Sequence Encoder

For each vessel $v_i$, the AIS stream at step $t$ is expressed in a scene-local metric frame and discretised into $1\,\mathrm{s}$ bins, producing a five-dimensional vector

$$a_{i,t} = [x_{i,t},\ y_{i,t},\ \mathrm{sog}_{i,t},\ \mathrm{coscog}_{i,t},\ \mathrm{sincog}_{i,t}] \in \mathbb{R}^5,$$

The sequence is processed by a single-layer, unidirectional GRU whose hidden dimension equals the shared model width (128 in the full-width model, 64 in Tiny):

$$h^{\mathrm{ais}}_{i,t} = \mathrm{GRU}(a_{i,1:t};\, \theta_{\mathrm{gru}}) \in \mathbb{R}^d,\ \ z^{\mathrm{ais}}_{i,t} = \mathrm{LN}\left(W_{\mathrm{ais}} h^{\mathrm{ais}}_{i,t} + b_{\mathrm{ais}}\right) \in \mathbb{R}^d.$$

A LayerNorm followed by a linear layer produces the AIS modality embedding $z^{\mathrm{ais}}_{i,t}$ at width $d$. The AIS encoder has 0.12 M parameters in the full-width model and 0.03 M in Tiny.

# 5.3.2 Radar Point Cloud Encoder

Each radar scan yields a point cloud $\mathcal{P}_{s,t} \in \mathbb{R}^{K\times 3}$ of $K$ local-$x$, local-$y$ and radial-Doppler returns; at most $K = 64$ points are retained per vessel and step. We adopt a PointNet++-style set abstraction with multi-scale grouping (MSG) at two radii $\{8, 25\}\,\mathrm{m}$ and 16 sampled neighbours per scale to encode each cloud:

$$h^{\mathrm{rad}}_{s,t} = \mathrm{PointNet}^{++}_{\mathrm{MSG}}(\mathcal{P}_{s,t};\, \theta_{\mathrm{pn}}) \in \mathbb{R}^{128}.$$

Radar returns are associated to vessels upstream, by gated nearest-neighbour matching against AIS-reported positions, so the encoder consumes a per-vessel cloud. Vessels with no AIS report inside the radar field of view retain the cloud centroid as their radar feature. The radar encoder has 0.08 M parameters (0.05 M in Tiny).

# 5.3.3 Vision Embedding via Detection and Re-Identification

For each camera, frames are processed by a YOLOv8-nano detector and an OSNet-x1.0 re-identification head, which together produce a per-detection appearance embedding. In the released configuration these two networks run upstream and the model consumes a 128-D normalised appearance embedding per vessel and step; a use_detector hook exists for calling those networks live, but it is disabled by default and the released configuration consumes the pre-computed embeddings. The per-vessel vision feature is the embedding associated with that vessel at that step:

$$h^{\mathrm{vis}}_{i,t} = \frac{1}{|\mathscr{F}_i|} \sum_{f\in\,\mathscr{F}_i} \mathrm{OSNet}\left(\mathrm{crop}_f\right) \in \mathbb{R}^{128}.$$

A learnable linear adapter projects the 128-D vision embedding to the shared hidden dimension $d$ before alignment. The vision adapter has 0.02 M parameters (0.01 M in Tiny); the YOLOv8 and OSNet weights sit upstream and are not part of the model total.

**Static vessel attributes.** Each vessel additionally carries a three-dimensional static descriptor (length, beam and a vessel-type code). A small MLP projects it to the shared width $d$ and the result is concatenated with the graph context to form the decoder memory; the branch contributes 0.01 M parameters.

# 5.4 Cross-Modal Temporal Alignment

AIS, radar, and vision streams are inherently asynchronous. In the release the simulator emits AIS reports at 2–10 s intervals and samples radar and camera observations on a 0.5 Hz grid, and all three are resampled onto a common 1 s grid. Within a single observation window the number of samples per modality therefore differs across vessels. To produce a synchronous embedding we apply DTW along the time axis, restricted to the sliding window. Given two modalities with timestamps $\{t^{\mathrm{ais}}_k\}$ and $\{t^{\mathrm{rad}}_k\}$, DTW finds the minimum-cost monotonic path:

$$\mathrm{DTW}(\mathcal{T}^{\,\mathrm{ais}}, \mathcal{T}^{\,\mathrm{rad}}) = \min_{\mathrm{path}\,P} \sum_{(p,q)\in\,P} \left\| \; x^{\mathrm{ais}}_{i,p} - x^{\mathrm{rad}}_{i,q} \right\|_2,$$

where $x^{\mathrm{ais}}$ and $x^{\mathrm{rad}}$ denote the raw scene-local positions of the two resampled streams: the alignment is computed at preprocessing time, before any encoder is applied (Section 5.2).

The optimal path pairs AIS samples with the temporally nearest radar returns. Entries missing along this path are flagged in the modality mask $m_i \in \{0,1\}^3$, whose components indicate whether the AIS, radar and vision inputs are observed at that step. The mask is passed to the downstream attention module, which fills the key and value rows of unavailable modalities with $-\infty$ before the softmax. A node is retained as long as at least one modality is alive, so an AIS dropout removes only the AIS channel rather than the whole vessel.

A learned 3-way cross-modal attention module then fuses the aligned embeddings:

$$\mathrm{Attn}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right)V,$$

with $Q = h_{i,T}^{\mathrm{ais}}$ (the query is the most recent AIS state), $K = V = [h_{i,T}^{\mathrm{ais}}; h_{i,T}^{\mathrm{rad}}; h_{i,T}^{\mathrm{vis}}]$, and the mask $m_i$ zeroing out unavailable modalities. The fused vessel representation is

$$h_i^{\mathrm{fuse}} = \mathrm{LayerNorm}\left(W_f \cdot \mathrm{Attn}(Q,K,V) + b_f\right) \in \mathbb{R}^{128}.$$

We deliberately use AIS as the query because it provides the most reliable vessel identity and timestamp, and we let attention learn the relative weight of radar and vision corrections. The alignment and fusion module adds 0.07 M parameters (0.02 M in Tiny).

# 5.5 Spatiotemporal Graph Construction

We construct a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ for each time step $t$ with three edge types:

- **Spatial edges** $\mathcal{E}^{\mathrm{spatial}}$: undirected edges between any two vessels whose planar distance is below $d_c = 150\,\mathrm{m}$. Self-loops are added for every node.
- **Temporal edges** $\mathcal{E}^{\mathrm{temporal}}$: directed edges from each node $v_i$ at step $t$ to the same node at step $t+1$ (and $t-1$), capturing identity continuity.
- **Shore edges** $\mathcal{E}^{\mathrm{shore}}$: undirected edges between each vessel node and its $k = 2$ nearest samples of the channel-bank polyline shipped with the scene, providing a "context" link to the channel geometry.

The combined adjacency tensor is $A_t = [A^{\mathrm{spatial}}; A^{\mathrm{temporal}}; A^{\mathrm{shore}}]$ with learnable edge-type embeddings added to the attention scores (Section 5.6). Collation caps the per-scene vessel count at $|\mathcal{V}_t| \le 24$ in the released configuration. Supplementary Fig. S4 illustrates the three edge types.

**Choice of the spatial threshold** $d_c$. The shipped configuration uses $d_c = 150\,\mathrm{m}$. At a typical sailing speed of about 5 kn ($\approx 2.6\,\mathrm{m/s}$) a vessel covers roughly 31 m over the 12 s prediction horizon, so 150 m spans several encounter envelopes while keeping the average spatial degree of the graph near unity

(1.11 at 150 m, Supplementary Table S4). Because the threshold is read at forward time rather than baked into the checkpoints, we measure its effect by sweeping $d_c$ at inference on the trained Tiny weights instead of re-training, and report the result in Supplementary Table S4: across 150–800 m the average spatial degree rises from 1.11 to 3.93 while ADE moves by only 0.14 m (18.47 m at 150 m, 18.61 m at 600 m). Neither failure regime one might expect (encounter-group fragmentation at small radii, interaction noise at large ones) is visible on this simulated benchmark, and a re-training sweep is left to future work: the flat response suggests the model is not sensitive enough to the threshold on this data for a re-training sweep to be informative.

# 5.6 Spatiotemporal Graph Attention Layer

The spatiotemporal graph attention (ST-GAT) layer is the core of Stage D. For each node $i$ at time $t$ we maintain a hidden state $h_{i,t} \in \mathbb{R}^d$. A single ST-GAT layer consists of two parallel attention mechanisms (spatial and temporal), whose outputs are concatenated, residual-added, and layer-normalised. We stack $L = 2$ such layers.

# 5.6.1 Spatial Attention

For a vessel node $v_i$ at time $t$, the spatial attention score to neighbour $v_j$ (connected by a spatial edge) is

$$e_{ij}^{s} = \mathrm{LeakyReLU}\left(\varvec{w}_{s}^{\top}[h_{i,t} \,||\, h_{j,t}]\right), \quad \varvec{w}_{s} \in \mathbb{R}^{2d},$$

where $||$ denotes concatenation and $\varvec{w}_{s}$ is a learned scoring vector. Scores are normalised over the neighbour set $\mathcal{N}(i)$:

$$\alpha_{ij}^{s} = \mathrm{softmax}_{j \in \mathcal{N}(i)}\left(e_{ij}^{s}\right).$$

The spatially attended representation is

$$h_{i,t}^{s} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{s} W_{v}^{(s)} h_{j,t}\right), \quad W_{v}^{(s)} \in \mathbb{R}^{d \times d},$$

with $W_{v}^{(s)}$ the per-head value projection.

# 5.6.2 Temporal Attention

For a vessel node $v_i$ at time $t$, the temporal attention score to the same vessel at times $\{t-1, t, t+1\}$ is

$$e_{i,k}^{\tau} = \mathrm{LeakyReLU}\left(\varvec{w}_{\tau}^{\top}[h_{i,t} \,||\, h_{i,k}]\right), \quad \varvec{w}_{\tau} \in \mathbb{R}^{2d}, \; k \in \{t-1, t, t+1\},$$

$$\alpha_{i,k}^{\tau} = \mathrm{softmax}_{k}\left(e_{i,k}^{\tau}\right),$$

$$h_{i,t}^{\tau} = \sigma\left(\sum_{k} \alpha_{i,k}^{\tau} W_v^{(\tau)} h_{i,k}\right), \; W_v^{(\tau)} \in \mathbb{R}^{d \times d}.$$

### 5.6.3 Combined Multi-Head Spatiotemporal Attention

We run $K_h$ attention heads in parallel for both mechanisms and concatenate; the full-width model uses $K_h = 4$ and the Tiny model $K_h = 2$:

$$h_{i,t}^{\mathrm{st}} = \Big\|_{k_h=1}^{K_h} \left[h_{i,t}^{s,k_h} + h_{i,t}^{\tau,k_h}\right] \in \mathbb{R}^{K_h d}.$$

A residual connection and layer normalisation give the final node embedding of a single layer:

$$h'_{i,t} = \mathrm{LayerNorm}\left(W_o \cdot \; h_{i,t}^{\mathrm{st}} + h_{i,t}\right) \in \mathbb{R}^{d}.$$

Parameter counts are taken from the released implementation rather than re-derived here, because the released layer adds per-edge-type value projections, gates, a position-wise feed-forward block and LayerNorm affine terms on top of the attention core: the ST-GAT stack (two layers, the shore-coordinate projection and the pooling projection) contributes 0.70 M parameters at $d = 128$ ( $L = 2$, $K_h = 4$) and 0.18 M at $d = 64$ ( $L = 2$, $K_h = 2$); the per-module breakdown is listed in Section 5.9.

The intuition behind this design is that spatial attention captures "which nearby vessels matter for predicting $v_i$'s next step", while temporal attention captures "which lags of $v_i$'s own trajectory matter". Their joint application reflects the empirical observation that inland vessels are strongly influenced by both neighbourhood and momentum.

## 5.7 Multi-Step Trajectory Decoder

The decoder generates $H$ future trajectory points conditioned on the sequence of graph-refined node embeddings $\{h'_{i,1:T}\}$. We use a Transformer decoder with $H$ query embeddings initialised from a sinusoidal positional code:

$$q_h = \mathrm{PE}\,(h)\,, \; h = 1, \ldots\,, H.$$

Each decoder block performs cross-attention from the trajectory queries to the graph encoder outputs, followed by a feed-forward network with hidden dimension $2d$ ( $\mathrm{dim}_{\mathrm{ff}} = 256$ at $d = 128$ and 128 at $d = 64$). The output of the final block is projected to a two-dimensional increment per future step:

$$\hat{y}_{i,h} = W_o \cdot \; \mathrm{DecoderBlock}(q_h, \{h'_{i,t}\})^{L_{\mathrm{dec}}},$$

where $\hat{y}_{i,h} = (\Delta x, \Delta y)_h$ is the predicted planar increment relative to the last observed position, and the absolute prediction is obtained by accumulating these increments. We deliberately predict increments rather than absolute coordinates because (i) the loss becomes scale-invariant and (ii) the network focuses on local motion dynamics rather than memorising channel geometry. The decoder has $L_{\mathrm{dec}} = 2$ blocks and 0.40 M parameters in the full-width model (0.10 M in Tiny).

## 5.7.1 Annealed Teacher Forcing

Feeding the decoder its own previous predictions from the first epoch makes an autoregressive trajectory decoder unstable, while never doing so at all lets it drift away from the manifold it is trained on. We therefore anneal the probability of replacing the model's own previous output with the ground-truth increment, linearly from $\epsilon_{\max} = 1.0$ to $\epsilon_{\min} = 0.2$ over $E_{\text{anneal}} = 60$ epochs:

$$\epsilon\left(\text{epoch}\right) = \epsilon_{\max} - \min\left(1, \frac{\text{epoch}}{E_{\text{anneal}}}\right)\left(\epsilon_{\max} - \epsilon_{\min}\right),$$

and hold it at $\epsilon = 0.2$ thereafter, so that the later part of training matches the inference condition. Across the released runs the best validation epoch varies widely with the variant and the seed — from epoch 5 to epoch 75 across the training runs of this study (the released checkpoint archive lists the per-run values) — which is consistent with the annealing phase containing most of the useful training on this dataset.

**Deterministic outputs and a probabilistic extension.** In its default configuration the decoder is deterministic: it returns a single most-likely increment per horizon step, which suits lock-scheduling and route-advisory tasks but yields no calibrated confidence for safety-critical warnings. The architecture can be extended to a lightweight probabilistic output by decoding $K$ candidate trajectories from $K$ parallel query embeddings, attaching a trainable scalar logit to each candidate, and training with a winner-take-all objective that back-propagates only through the candidate closest to the ground truth; the softmax over the logits then provides a scene-conditional probability mass that a VTS operator could threshold by risk rather than by raw displacement error. We keep this extension for future work (Section 4.3) and analyse the failure modes of the deterministic setting in the Limitations (Section 4.2).

# 5.8 Loss Function

The model is trained with a weighted combination of three losses:

$$\mathscr{L} = \lambda_1 \mathscr{L}_{\text{ADE}} + \lambda_2 \mathscr{L}_{\text{FDE}} + \lambda_3 \mathscr{L}_{\text{heading}}.$$

**Average Displacement Error (ADE)**:

$$\mathscr{L}_{\text{ADE}} = \frac{1}{H} \sum_{h=1}^{H} \left\| \text{pos}_{i,h} - \text{pos}_{i,h}^{*} \right\|_2.$$

**Final Displacement Error (FDE)**:

$$\mathscr{L}_{\text{FDE}} = \left\| \text{pos}_{i,H} - \text{pos}_{i,H}^{*} \right\|_2.$$

**Heading Cosine Loss**

the heading term penalises the direction of the predicted increment at every valid future step, so it constrains the velocity vector's orientation only

$$\mathscr{L}_{\text{heading}} = \frac{1}{|\mathscr{H}_i|} \sum_{h \in \mathscr{H}_i} \left( 1 - \frac{\widehat{u}_{i,h} \cdot u^*_{i,h}}{\| \widehat{u}_{i,h} \| \, \| u^*_{i,h} \|} \right),$$

where $\widehat{u}_{i,h}$ and $u^*_{i,h}$ are the predicted and ground-truth increments at step $h$, and $\mathscr{H}_i$ is the set of valid future steps for vessel $i$.

The released configuration uses $\lambda_1 = 1.0$, $\lambda_2 = 1.0$ and $\lambda_3 = 0.2$, keeping the heading term a regulariser rather than a co-equal objective.

# 5.9 Implementation Notes

All experiments run on CPU with PyTorch 2.x in single precision; no GPU is required to reproduce any number in this paper. Training uses AdamW with $\mathrm{lr} = 1 \times 10^{-3}$ and a step schedule at epochs 40 and 70, weight decay $10^{-4}$, batch size 16, three warm-up epochs and gradient clipping at 5.0. The full-width model has 1.47 M trainable parameters (AIS 0.12 M + radar 0.08 M + vision adapter 0.02 M + modality projections 0.05 M + alignment and fusion 0.07 M + ST-GAT 0.70 M + static and shore context 0.04 M + decoder 0.40 M); the Tiny configuration has 0.40 M. Training to early stopping takes 884 s (15 min) for the full-width model and 1,100 s (18 min) for Tiny on seed 0, the smaller model training longer because it selects a later best epoch on that seed; inference throughput is 179 vessels/s for Tiny and 105 vessels/s for the full-width model. The YOLOv8-nano and OSNet weights used upstream by the vision branch are not included in these counts, because the released vision adapter consumes pre-computed embeddings. Source code, the simulator, and the field-recording data layout are released with this paper.

# 5.10 Evaluation Metrics

We report five quantities spanning accuracy, geometric fidelity and deployment efficiency:

1. **ADE** (Average Displacement Error, metres): mean Euclidean error over the predicted horizon.
2. **FDE** (Final Displacement Error, metres): error at the final predicted step.
3. **MR@2m** (Miss Rate, %): fraction of trajectories whose endpoint error exceeds 2 m.
4. **HE** (Heading Error, degrees): mean angle between the predicted and the ground-truth velocity direction over the forecast horizon.
5. **Throughput** (vessels per second): test vessels processed per second during the evaluation pass on CPU.

Lower is better for ADE, FDE, MR and HE; higher is better for throughput. One caveat applies to MR@2m: on this testbed every method has an FDE of roughly 30 m, so the 2 m threshold is exceeded by essentially every trajectory and the metric saturates near 100% (Table 2). We report it for consistency with the released metric implementation, but the ranking in this section is carried by ADE and FDE. A second caveat applies to HE: it is evaluated on the per-step displacement directions of the (noisy) simulated tracks rather than on smoothed headings, which inflates its absolute value for every method alike; the parameter-free constant-velocity row of Table 2 serves as the reference point for interpreting it.

# 5.11 Baseline Methods

Three baselines can be trained inside the released repository on identical splits with identical preprocessing: a Bi-LSTM encoder over the AIS sequence alone (hidden width 64), Social-STGCNN [31], also AIS-only, and an AIS-only Transformer encoder. All use the same schedule and batch size as our model.

**AIS-only Transformer.** The Transformer baseline consumes the same 5-D kinematic window as the Bi-LSTM — no radar, no vision, no graph — and maps the pooled encoder state to the 12-step displacement through the same decoder head, so it isolates the attention architecture from the multi-source machinery and provides a like-for-like check on the fused model's own attention stack. Because a Transformer's capacity is not recoverable from its weights alone, its width, depth, head count and feed-forward size are written to a baseline_arch.json sidecar beside the checkpoint at training time, and the released tools/evaluate.py and tools/predict.py read that sidecar when reloading it. Section 3.2 reports the same architecture at four capacities spanning 78 K to 4.79 M parameters.

**Constant-velocity (CV) baseline.** Trajectory-prediction studies conventionally anchor the absolute error scale with a parameter-free constant-velocity extrapolation, and we do the same: for each vessel the per-step displacement is estimated from its last two observed AIS fixes on the 1 s scene grid and held constant over the 12-step horizon. Vessels with no AIS fix in the window are not evaluable (38 of 228 test vessels), exactly as for any AIS-only predictor. The CV row of Table 2 and the paired statistics of Section 3.2 are produced by tools/eval_cv_stats.py in the released package, using the same metric implementation as the trained models.

The wider comparison set familiar from the trajectory-prediction literature (Trajectron++, mmTransformer, BEVFusion, GMAN, MultiPath++) is not reproduced here. Those systems were evaluated on other datasets with other horizons and metric conventions, and none of them consumes an AIS-radar-vision inland stream without substantial re-engineering; quoting their published numbers next to ours would compare incomparable quantities. We therefore restrict Table 2 to models that were actually run on this testbed, and list extending the baseline set as future work.

## Declarations

# Competing Interests

The authors declare no competing interests.

# Author Contribution

All authors designed the study and approved the submitted version. J.Z. implemented the simulator, the reference model and the training pipeline, and ran the main comparison, the capacity ladder and the three controls of Section 3.2; Y.L. designed the evaluation protocol and the scene-level statistical analysis; J.L. supervised the project; J.Z. wrote the first draft and all authors revised the manuscript. Contributions follow the CRediT taxonomy: conceptualisation, methodology, software, validation, formal analysis, investigation, data curation, writing — original draft, writing — review and editing, visualisation, supervision.

# Acknowledgement

The authors received no specific funding for this work. The authors thank colleagues at their institutions for comments on an earlier draft of the manuscript, and acknowledge the open-source projects the released pipeline builds on (PyTorch, NumPy, SciPy, Matplotlib, PyYAML). All training and evaluation were carried out on CPU.

# Data Availability

All data generated and analysed in this study are openly available. The InlandVessel-2K-Sim scene data (the 160-scene released split, its unaligned variant, the controlled noisy/clean observation-noise pair, the two smoke sets, and the exact train/validation/test scene lists used by every reported result), the trained checkpoints for all 57 runs, and the reference code package are archived in a single Zenodo deposit at https://doi.org/10.5281/zenodo.23175772. Data and checkpoints are licensed CC BY 4.0 and the code MIT. Each archive carries a SHA256SUMS.txt, and the checkpoint archive carries a checkpoint_index.csv mapping every run directory onto the table it supports. The code package contains the model, the training and evaluation pipeline, the simulator, a one-command driver (tools/run_all.py) that regenerates every table in this paper and the supplementary material, and the drivers of the three controls (tools/run_datascale.py, tools/run_noise_probe.py, tools/run_m6_transformer.py). One caveat on regenerating the released split: the build that produced the shipped .npz files seeded each channel from hash(waterway), and Python randomises string hashing per process, so the shipped files are not bit-reproducible by the current generator; they are released unmodified because every reported number is computed on them, while the current generator seeds deterministically from --seed and a channel index. No field recordings were used in this study: every reported number is computed on simulated data generated by the released simulator. The supplementary material provides Tables S1-S5, the architecture diagrams (Supplementary Figures S1-S4), the protocol descriptions, the reproduction commands and a statement of provenance for every reported number.

## References

1. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).

2. Graves, A. & Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **18**, 602–610 (2005).
3. Vaswani, A. et al. Attention is all you need. In Proc. *Adv. Neural Inf. Process. Syst.* 5998–6008 (2017).
4. Salzmann, T., Ivanovic, B., Chakravarty, P. & Pavone, M. Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. Preprint at (2020). https://arxiv.org/abs/2001.03093
5. Yu, B., Yin, H. & Zhu, Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In Proc. *Int. Joint Conf. Artif. Intell. (IJCAI)* 3634–3640 (2018).
6. Qi, C. R., Yi, L., Su, H. & Guibas, L. J. PointNet++: deep hierarchical feature learning on point sets in a metric space. In Proc. *Adv. Neural Inf. Process. Syst.* 5099–5108 (2017).
7. Jocher, G., Chaurasia, A. & Qiu *J. Ultralytics* YOLOv8. GitHub (2023). https://github.com/ultralytics/ultralytics
8. Zhou, K., Yang, Y., Cavallaro, A. & Xiang, T. Omni-scale feature learning for person re-identification. In Proc. *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)* 3702–3712 (2019).
9. Li, J. et al. Active warning system for bridge collision prevention: design and application. In Proc. *Int. Conf. Transp. Inf. Saf. (ICTIS)* 1–8 (2025).
10. Liu, Z. et al. BEVFusion: multi-task multi-sensor fusion with unified bird's-eye view representation. In Proc. *IEEE Int. Conf. Robot. Autom. (ICRA)* 11112–11119 (2023).
11. Bai, X. et al. TransFusion: robust LiDAR-camera fusion for 3D object detection with transformers. In Proc. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* 1080–1089 (2022).
12. Capobianco, S., Millefiori, L. M., Forti, N., Braca, P. & Willett, P. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **57**, 4329–4346 (2021).
13. Alahi, A. et al. Social LSTM: human trajectory prediction in crowded spaces. In Proc. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* 961–971 (2016).
14. Forti, N., Millefiori, L. M., Braca, P. & Willett, P. Prediction of vessel trajectories from AIS data via sequence-to-sequence recurrent neural networks. In Proc. *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)* 8936–8940 (2020).
15. Nguyen, D. & Fablet, R. A transformer network with sparse augmented data representation and cross entropy loss for AIS-based vessel trajectory prediction. *IEEE Access.* **12**, 21596–21609 (2024).
16. Nguyen, D., Vadaine, R., Hajduch, G., Garello, R. & Fablet, R. GeoTrackNet—a maritime anomaly detector using probabilistic neural network representation of AIS tracks and a contrario detection. *IEEE Trans. Intell. Transp. Syst.* **23**, 5655–5667 (2022).
17. Shin, Y. et al. Deep learning framework for vessel trajectory prediction using auxiliary tasks and convolutional networks. *Eng. Appl. Artif. Intell.* **132**, 107936 (2024).
18. Bi, J., Cheng, H., Zhang, W., Bao, K. & Wang, P. Artificial intelligence in ship trajectory prediction. *J. Mar. Sci. Eng.* **12**, 769 (2024).

19. Guo, S. & Ma, W. A systematic review of deep learning-based methods for ship trajectory prediction. *J. Mar. Sci. Eng.* **14**, 810 (2026).
20. Wan, J., Wen, J. & Wu, Y. Inland river ship navigation situation monitoring system based on multi-source data fusion. In Proc. *Int. Conf. Transp. Inf. Saf. (ICTIS)* 1–8 (2025).
21. Liu, J., Zhu, K., Shu, Y., Zhang, Y. & Zheng, P. Trajectory prediction model of a ship crossing under a bridge based on multi-head attention mechanism and Wasserstein GAN with gradient penalty. *Ocean Eng* **121551** (2025).
22. Wang, X., Zheng, Y., Fang, Z., Liu, Z. & Liu, Y. A novel ship trajectory prediction method based on E-Informer. *IET Conf. Proc.* 467–472 (2025). (2024).
23. Zhang, Z., Yang, N. & Yang, Y. Autonomous navigation and collision prediction of port channel based on computer vision and lidar. *Sci. Rep.* **14**, 11300 (2024).
24. Shao, Z. et al. An efficient model for small object detection in the maritime environment. *Appl Ocean. Res* 104194 (2024).
25. Xiao, Y., Hu, Y., Liu, J., Xiao, Y. & Liu, Q. An adaptive multimodal data vessel trajectory prediction model based on a satellite automatic identification system and environmental data. *J. Mar. Sci. Eng.* **12**, 513 (2024).
26. Wu, Z., Pan, S., Long, G., Jiang, J. & Zhang, C. Graph WaveNet for deep spatial-temporal graph modeling. In Proc. *Int. Joint Conf. Artif. Intell. (IJCAI)* 1907–1913 (2019).
27. Zheng, C., Fan, X., Wang, C. & Qi, J. GMAN: a graph multi-attention network for traffic prediction. In Proc. *AAAI Conf. Artif. Intell. (AAAI)* 1234–1241 (2020).
28. Liu, Y., Zhang, J., Fang, L., Jiang, Q. & Zhou, B. Multimodal motion prediction with stacked transformers. In Proc. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* 5309–5319 (2021).
29. Veličković, P. et al. Graph attention networks. Preprint at (2018). https://arxiv.org/abs/1710.10903
30. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. Preprint at (2017). https://arxiv.org/abs/1609.02907
31. Mohamed, A., Qian, K., Elhoseiny, M. & Claudel, C. Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory forecasting. In Proc. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* 5123–5132 (2020).
32. Huang, Y., Bi, H., Li, Z., Mao, T. & Wang, Z. STGAT: modeling spatial-temporal interactions for human trajectory prediction. In Proc. *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)* 6272–6281 (2019).
33. Feng, H., Cao, G., Xu, H. & Ge, S. S. IS-STGCNN: an improved social spatial-temporal graph convolutional neural network for ship trajectory prediction. *Ocean. Eng.* **266**, 112960 (2022).
34. Varadarajan, B. et al. MultiPath++: efficient information fusion and trajectory aggregation for behavior prediction. Preprint at (2022). https://arxiv.org/abs/2111.14973
35. Zhou, Z., Ye, L., Wang, J., Wu, K. & Lu, K. HiVT: hierarchical vector transformer for multi-agent motion prediction. In Proc. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)* 8823–8833 (2022).

36. Nayakanti, N. et al. Wayformer: motion forecasting via simple and efficient attention networks. In Proc. *IEEE Int. Conf. Robot. Autom. (ICRA)* 2980–2987 (2023).

37. Berndt, D. J. & Clifford, J. Using dynamic time warping to find patterns in time series. In Proc. *AAAI-94 Workshop Knowl. Discov. Databases (KDD)* 359–370 (1994).

38. Singh, S. K., Fowdur, J. S., Gawlikowski, J. & Medina, D. Leveraging graph and deep learning uncertainties to detect anomalous maritime trajectories. *IEEE Trans. Intell. Transp. Syst.* **23**, 23488–23502 (2022).

# Figures

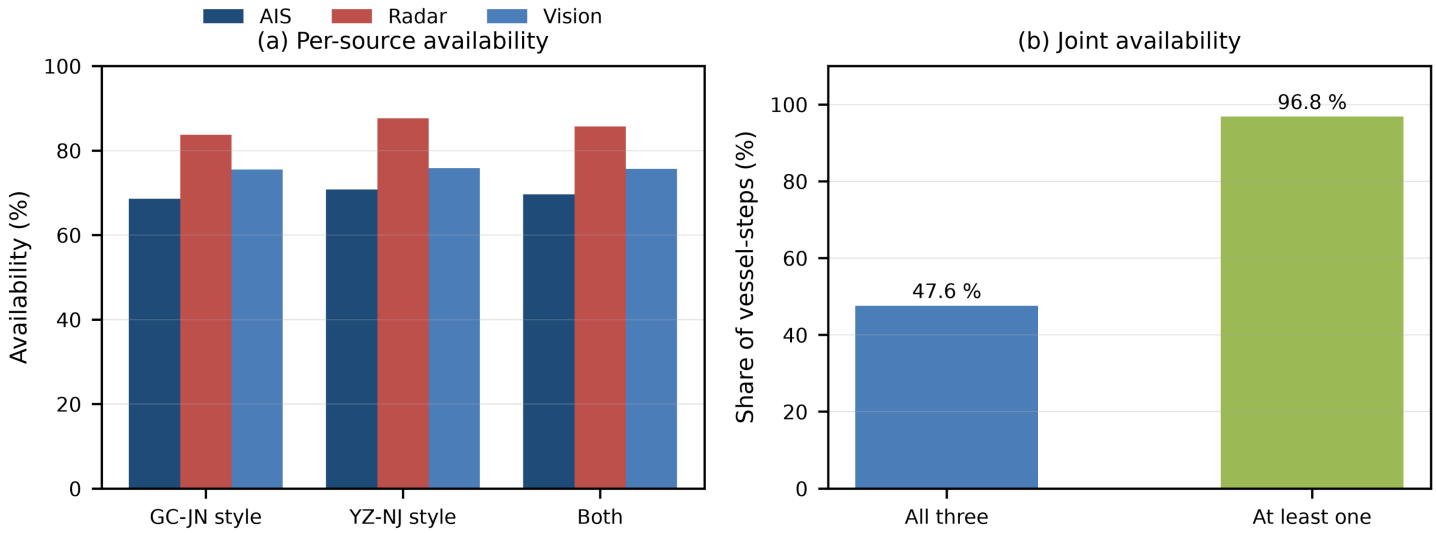



**Figure 1**

Unnumbered image in the 3. Results section.

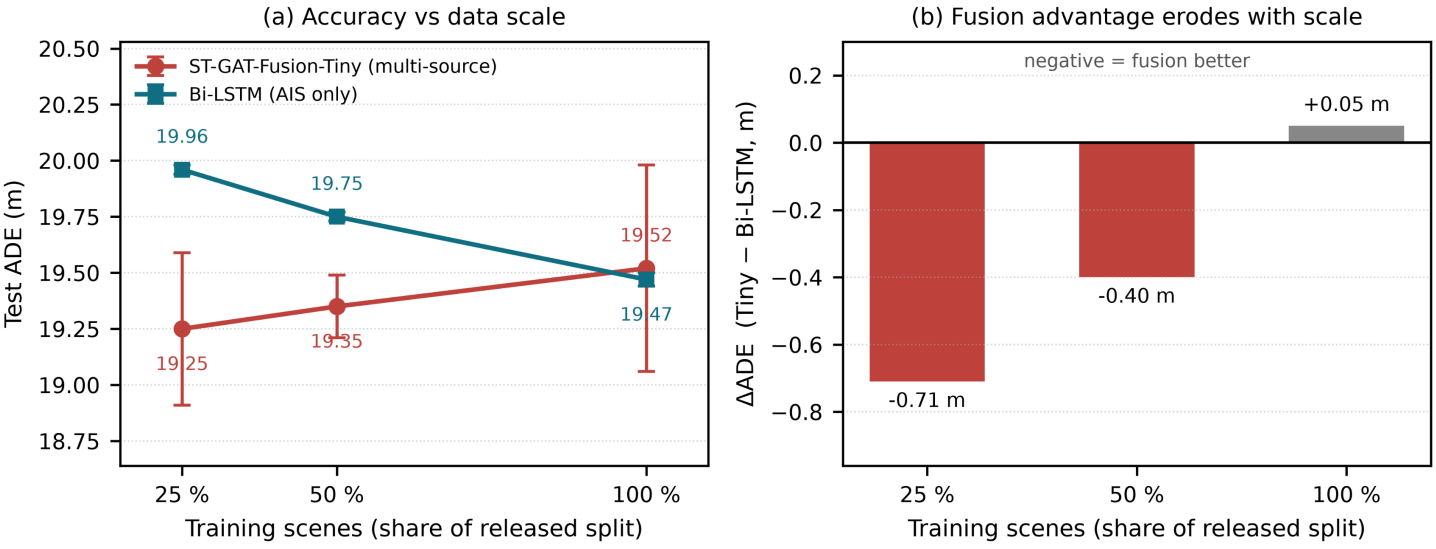


**Figure 2**

Unnumbered image in the 3. Results section.

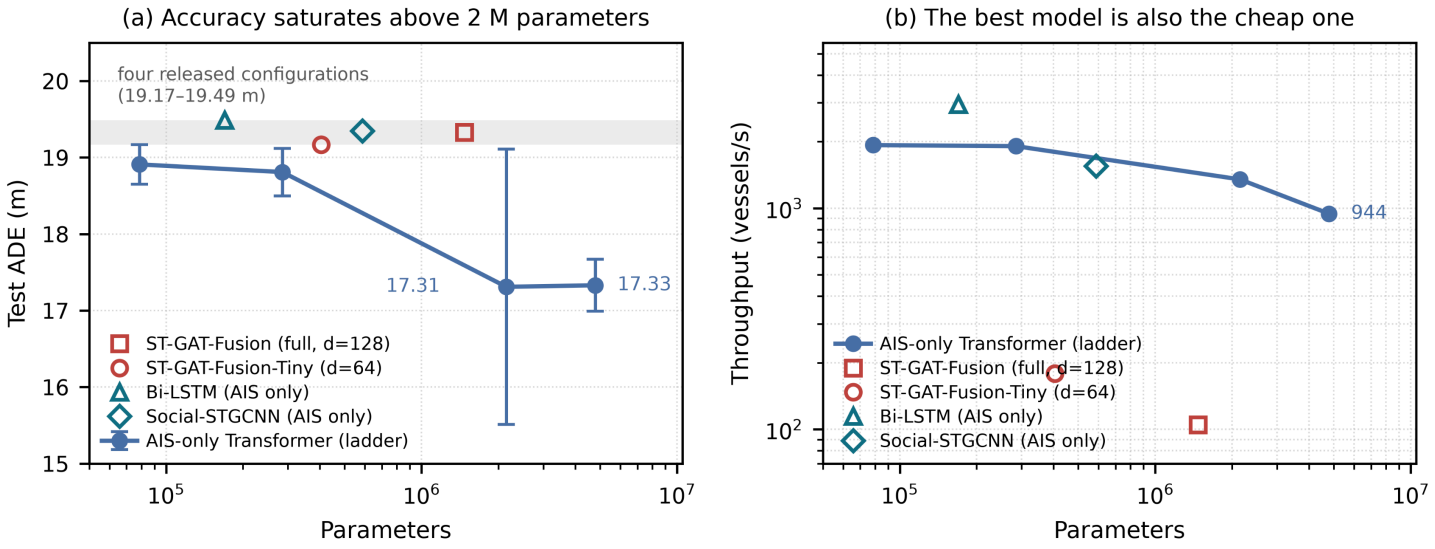



**Figure 3**

Unnumbered image in the 3. Results section.

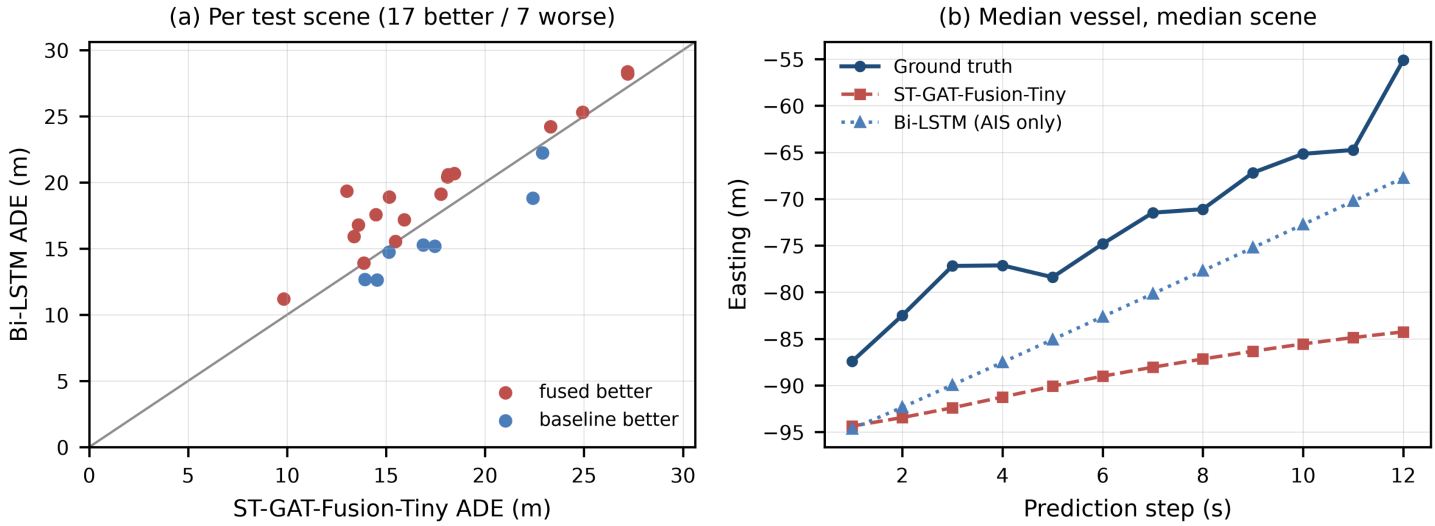



**Figure 4**

Unnumbered image in the 3. Results section.

# Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- STGATFusionSupplementaryMaterial.docx