Non-Parametric Spatiotemporal Trajectory Prediction via State-Conditioned Transition Sampling

Michael Fore
mikefore@amazon.com
Amazon

Akshay Jain
jainaaks@amazon.com
Amazon

Duncan Botti
dunbotti@amazon.com
Amazon

Newel Hirst
nhirst@amazon.com
Amazon

Justin Downes
jusdow@amazon.com
Amazon

Rohan Pradhan
awsrohan@amazon.com
Amazon

Sharlina Keshava
skeshava@amazon.com
Amazon

ABSTRACT

We present a nonparametric method for multi-modal trajectory prediction that requires no GPU, fits in seconds on CPU, and matches or exceeds a 57M-parameter transformer. The method builds a transition table of historical state-to-next-position pairs and retrieves neighbors using a product kernel over spatial proximity, bearing, speed, and temporal context. Two inference modes operate over this shared representation: diversity-penalized sampling produces multiple trajectories covering distinct plausible routes, while beam search finds the single highest-likelihood path. On the TrAISformer benchmark (Danish Maritime Authority AIS, 1,481 test trajectories), our beam search achieves 9.13 km top-1 ADE at 3 hours (vs. 9.51 km for TrAISformer argmax) and our diverse sampling achieves 2.38 km best-of-16 ADE (vs. 2.80 km for TrAISformer sampling). In data-scarce regimes, our method outperforms the transformer by 2–4× and remains stable down to 2% of training data where TrAISformer degrades catastrophically. The method requires zero learned parameters, works immediately on new regions given sufficient historical data, and provides interpretable predictions grounded in observed transitions.

1 INTRODUCTION

Predicting the future positions of moving entities—vessels, vehicles, pedestrians—is a core problem in spatial computing with applications in maritime surveillance, traffic management, and search-and-rescue operations. The task is inherently multi-modal: at route junctions, a vessel may proceed in any of several directions, and a useful predictor must represent this uncertainty rather than committing to a single path.

Recent deep learning approaches frame trajectory prediction as sequence modeling. TrAISformer [6] discretizes AIS (Automatic Identification System) positions into grid cells and applies a transformer architecture to predict future cell sequences, achieving state-of-the-art results on maritime trajectory benchmarks. However, such models require GPU training (57.4M parameters), degrade with limited training data, and must be retrained for new geographic regions or entity types.

Classical nonparametric methods avoid these costs. Hexeberg et al. [3] proposed Single Point Neighbor Search (SPNS): find the nearest historical AIS observation to the current vessel position and use its subsequent position as the prediction. This requires no training but uses only spatial proximity for neighbor selection, ignoring heading, speed, and temporal context. As a result, it performs poorly in multi-route environments where vessels at the same location may be traveling in different directions.

We present a method that bridges this gap. Our approach builds a transition table of historical state-to-next-position pairs and retrieves neighbors using a product kernel over spatial proximity, bearing (von Mises), speed (Gaussian), and time-of-day/day-of-week (cyclic Gaussian). We introduce two inference modes over this shared representation:

(1) **Diverse sampling** with a spatial repulsion penalty that forces successive trajectory samples to explore different routes, producing calibrated multi-modal predictions.
(2) **Beam search** that maintains multiple candidate paths and prunes by cumulative path likelihood, producing accurate single-trajectory predictions.

On the TrAISformer benchmark (DMA Danish maritime AIS, 1,481 test trajectories, 3-hour prediction horizon), our method achieves:

- Top-1 ADE: 9.13 km (beam search) vs. 9.51 km (TrAISformer argmax), a 4% improvement
- Best-of-16 ADE: 2.38 km (diverse sampling) vs. 2.80 km (TrAISformer sampling), a 15% improvement

while requiring no GPU, no learned parameters, and only seconds of CPU setup. The method is effective in sparse data scenarios where the transformer degrades catastrophically, by maintaining reasonable accuracy using only 2% of training data.

Our contributions are:

(1) A state-conditioned transition predictor using product-kernel weighting that extends spatial-only neighbor search [3] with bearing, speed, and temporal conditioning.
(2) Diversity-promoting sequential sampling [9] applied to spatial transition lookup, producing multi-modal trajectory predictions that cover distinct plausible routes.
(3) Systematic evaluation demonstrating that this training-free approach matches or exceeds a 57M-parameter transformer across multiple metrics, datasets, and data-availability regimes.

2 METHOD

Our method operates in two phases. First, in an offline setup step, we construct a transition table with spatial indices. Second, we leverage the transition table in online inference via one of two modes: diverse sampling or beam search. Both of these inference modes share the same weighted neighbor lookup created in the first stage.

2.1 Transition Table

From training trajectories sampled at regular intervals (10 minutes in our experiments), we extract all pairs of observations separated by a fixed prediction step Δt (1 hour). Each entry stores the current state $\mathbf{s}_i = (\mathrm{lat}_i, \mathrm{lon}_i, v_i, \theta_i, h_i, d_i)$—position, speed, bearing, hour-of-day, and day-of-week—along with the next position $\mathbf{p}_i^+ = (\mathrm{lat}_i^+, \mathrm{lon}_i^+)$ observed Δt later. We build a BallTree spatial index on the current positions for efficient neighbor retrieval. Construction takes approximately 2 seconds on CPU for 150,000 transitions.

2.2 State-Conditioned Weighted Lookup

Given a query state $\mathbf{q} = (\mathrm{lat}, \mathrm{lon}, v, \theta, h, d)$ representing position, speed (knots), bearing (degrees), hour-of-day, and day-of-week, we retrieve all transition entries within an adaptive spatial bandwidth and assign each entry i a weight that is the product of five kernel terms:

$$w_i = K_{\mathrm{sp}} \cdot K_{\mathrm{brg}} \cdot K_{\mathrm{spd}} \cdot K_{\mathrm{hr}} \cdot K_{\mathrm{dow}} \tag{1}$$

where:

- K_{sp}: quartic kernel with adaptive bandwidth (following Abramson's square-root rule [1]), applied to great-circle distance between query and entry positions.
- $K_{\mathrm{brg}} = f_{\mathrm{vM}}(\theta - \theta_i; \kappa)$: von Mises density with concentration $\kappa = 16$, giving effective weight within $\approx \pm 20$ of the query bearing.
- $K_{\mathrm{spd}} = \exp\left(-\frac{(v-v_i)^2}{2\sigma_v^2}\right)$ with $\sigma_v = 1$ knot.
- $K_{\mathrm{hr}} = \exp\left(-\frac{\Delta_{\mathrm{circ}}(h,h_i;24)^2}{2\sigma_h^2}\right)$ with $\sigma_h = 6$ hours.
- $K_{\mathrm{dow}} = \exp\left(-\frac{\Delta_{\mathrm{circ}}(d,d_i;7)^2}{2\sigma_d^2}\right)$ with $\sigma_d = 2$ days.

Here $\Delta_{\mathrm{circ}}(a, b;\ P) = \min(|a-b|,\ P-|a-b|)$ is the circular distance with period P. This ensures that, for example, using a 24-hour period, hour 23 and hour 1 are two hours apart rather than twenty-two.

The result is a weighted set of next-positions $\{(\mathbf{p}_i^+, w_i)\}$ representing where entities in similar states went next. This is a form of Nadaraya-Watson [5, 10] kernel regression applied to state-conditioned spatial transitions.

2.3 Diverse Sampling

To produce N trajectory branches to collectively cover plausible future routes, we sample branches sequentially with a spatial repulsion penalty, analogous to the diversity-augmented objective in Diverse Beam Search [9] but applied at the observation level in continuous space rather than at the token level in discrete sequences.

For branch $j = 1, \ldots, N$, at each time step t:

(1) Compute weights $\{w_i\}$ via the product kernel (Section 2.2).

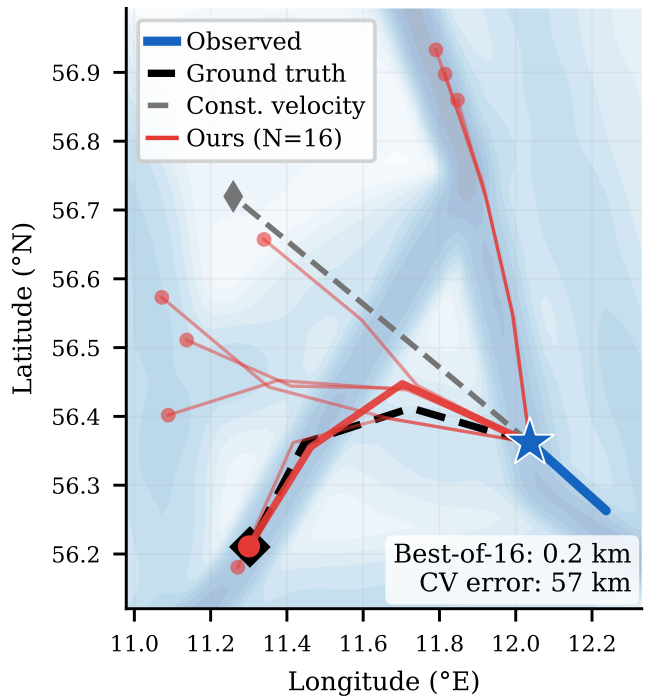

Figure 1: Our method produces diverse predictions that cover multiple plausible shipping corridors for one vessel in the DMA dataset. In the background behind the routes, the figure displays a spatial-only KDE fit on AIS traffic in this area. The KDE is intended to provide the reader visual context for comparing the various projected routes - the KDE is not itself incorporated into the trajectory prediction method.

(2) For $j > 1$, apply a diversity penalty to each observation's weight:

$$w_i' = w_i \cdot \prod_{k=1}^{j-1} \left(1 - \exp\left(-\frac{\|\mathbf{p}_i^+ - \mathbf{p}_k^{(t)}\|^2}{2\sigma_{\mathrm{div}}^2}\right)\right) \tag{2}$$

where $\mathbf{p}_k^{(t)}$ is the position of previously generated branch k at step t, and $\sigma_{\mathrm{div}} = 5$ km. For each observation i, if its next-position $\mathbf{p}_i^+$ is near any previous branch, the penalty term approaches zero and suppresses that observation. This forces the new branch to sample from observations leading to different regions.

(3) Sample one observation from the modified weight distribution.

(4) Smooth: compute the weighted mean of all next-positions within ±15 of the sampled bearing direction.

(5) Update the query state from the smoothed prediction and proceed to step $t + 1$.

At each time-step, each branch makes independent stochastic decisions conditioned on the diversity-modified distribution. Once branches are fully generated, we rank them by their unpenalized cumulative log-density (essentially removing the diversity penalty for final ranking).

2.4 Beam Search

As an alternate variant, we implement beam search to find the highest-likelihood complete path. At each time step, for each of B active branches: (1) generate C candidate next-positions by sampling from the product-kernel weights (no diversity penalty); (2) pool all $B \times C$ candidates and retain the top B by cumulative log-density. Paths that remain on high-traffic corridors accumulate high scores; paths that wander off-route are pruned. In our experiments, we use $B = 16$ and $C = 3$. The final ranking uses cumulative log density like our diverse sampling method.

2.5 Autoregressive Propagation

Both modes are autoregressive: at each step, the predicted position and bearing become the query state for the next step. This allows trajectories to follow curves and respond to changing spatial context. The prediction horizon is H steps of Δt each. In our experiments, we use $\Delta_t = 1$–hour and predict at both a 1 hour ($H = 1$) and 3-hour ($H = 3$) horizon.

3 EXPERIMENTS

3.1 Setup

Dataset. We evaluate on the DMA AIS dataset (Danish Maritime Authority): vessel traffic in the Kattegat/Danish Straits region, sampled at 10-minute intervals. We follow TrAISformer's reference split: 10,605 training trajectories (January–December 2019 excluding the test period, 2,517 vessels) and 1,481 test trajectories (March 21–31, 2019, with no overlap with training). Each trajectory has a 3-hour observation window followed by a 3-hour prediction horizon.

Baselines.

- *Constant velocity (CV)*: Linear extrapolation from the last observed position and bearing.
- *Hexeberg SPNS* [3]: Spatial-only neighbor lookup with our diversity penalty added ($N = 16$, $\sigma_{\text{div}} = 5$ km) to enable best-of-N evaluation.
- *LSTM seq2seq*: 2-layer encoder-decoder LSTM (401,924 parameters), trained for 50 epochs with MSE loss. Best-of-16 via Gaussian noise injection on the hidden state.
- *TrAISformer* [6]: 57.4M-parameter transformer trained on the same data for 50 epochs on a T4 GPU. Top-1 via argmax decoding; best-of-16 via categorical sampling.

We report TrAISformer results from our own reproduction using their public code and data, not from their paper directly. However, our reproduction achieves slightly *better* results than reported in the original paper (e.g., 1.51 nmi at 3h best-of-16 vs. their reported 1.64 nmi), likely due to minor differences in hyperparameters or other training choices. All methods in Table 1 are evaluated on identical test trajectories.

Metrics. We report Average Displacement Error (ADE) in kilometers at 1-hour and 3-hour horizons. We report *top-1* (error of the single best-ranked prediction). However, trajectory prediction scenarios like DMA are inherently multi-modal - there are likely to be common routes that share a trajectory for some time period then fork into different directions. As such, we follow standard practice in multi-modal trajectory prediction by also reporting *best-of-N* (minimum error across $N = 16$ predictions) [2]. The practice of reporting best-of-16 was followed by TrAISformer as well. To assess probabilistic calibration, we additionally report the Continuous Ranked Probability Score (CRPS) in its energy form: $\text{CRPS} = \mathbb{E}[\|X - y\|] - \frac{1}{2}\mathbb{E}[\|X - X'\|]$, where X, X' are samples and y is the ground truth. Lower CRPS indicates samples that are both accurate and well-concentrated.

Table 1: Trajectory prediction on DMA AIS (1,481 test trajectories, 3-hour prediction). ADE in km.

Method	Params	1h top-1	3h top-1	3h best-16	3h CRPS
Const. velocity	0	3.66	19.29	–	–
Hexeberg SPNS	0	15.47	43.59	8.91	37.99
LSTM seq2seq	402K	3.12	9.05	4.99	10.86
TrAISformer	57.4M	2.18	9.51	2.80	**8.23**
Ours (beam)	0	2.31	**9.13**	7.11	8.44
Ours (diverse)	0	4.02	13.39	**2.38**	13.74

Parameters. Transition table with 150,000 entries subsampled from training data. Beam search: $B = 16$, $C = 3$. Diverse sampling: $N = 16$, $\sigma_{\text{div}} = 5$ km, with temporal weighting ($\sigma_h = 6$h, $\sigma_d = 2$d).

3.2 Main Results

Table 1 presents the full comparison on the DMA test set.

Our beam search achieves the best 3-hour top-1 ADE (9.13 km vs. TrAISformer's 9.51 km), while our diverse sampling achieves the best 3-hour best-of-16 ADE (2.38 km vs. 2.80 km). TrAISformer retains an advantage at the 1-hour top-1 metric (2.18 vs. 2.31 km), where its attention over the 3-hour observation window helps disambiguate routes at junctions.

The Hexeberg baseline (spatial-only weighting) achieves 8.91 km best-of-16—adding bearing, speed, and temporal conditioning reduces this to 2.38 km, a 73% improvement that quantifies the value of multi-dimensional kernel conditioning. On CRPS, our beam search (8.44 km) is comparable to TrAISformer (8.23 km), as both produce concentrated samples along the most likely route. Our diverse mode has higher CRPS (13.74 km) due to the diversity penalty spreading samples across multiple routes—the same property that yields its best-of-16 advantage.

3.3 Data Efficiency

We evaluate how both methods degrade as training data is reduced. For these experiments, we train on subsets of January–November 2019 and test on December (290 trajectories). For our method only, the transition table is capped at 150,000 entries for efficiency; this cap is only binding above ∼25% of the full dataset, meaning performance at 50% and 100% uses the same number of transitions. The TrAISformer baselines use the full dataset at whatever percentage we are experimenting at with no cap (i.e. 100% uses all training samples). Results are shown in Table 2.

Our method's performance is stable from 100% down to 5% of training data (2.32–2.45 km), degrading only at 1% (3.54 km). TrAISformer degrades from 2.80 km at full data to 14.87 km at 1%—a 5.3× increase. At 10% of data (approximately 5 weeks of AIS collection), our method is 2.2× more accurate than TrAISformer.

Table 2: Data efficiency (3h best-of-16 ADE in km, December test).

Training data	Ours ↓	TrAISformer ↓	Ratio
100%	2.32	2.80	0.83×
50%	2.24	2.85	0.79×
10%	2.34	5.16	0.45×
5%	2.45	7.14	0.34×
1%	3.54	14.87	0.24×

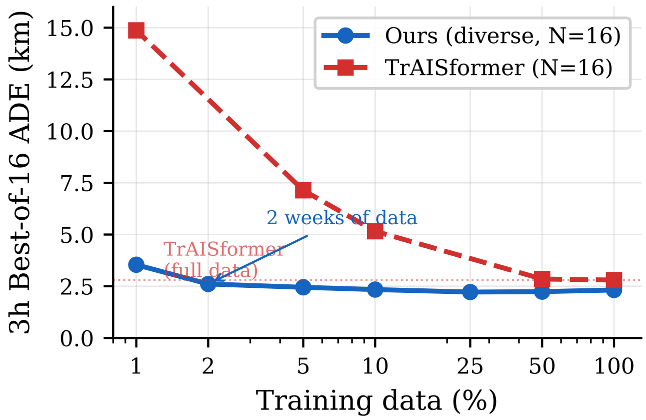

Figure 2: Data efficiency: 3-hour best-of-16 ADE vs. training data fraction. Our method (blue) is stable across the range; TrAISformer (red) degrades exponentially below 50%.

3.4 Computational Requirements

Our method requires no training loop and runs entirely on CPU. Setup (building the BallTree index over 150,000 transitions) and inference both complete in seconds. TrAISformer requires GPU training (NVIDIA T4, ~33 minutes) and its inference on CPU is approximately 20× slower than ours for the same test set, though we note this comparison is implementation-dependent.

4 DISCUSSION AND CONCLUSION

TrAISformer generates 16 branches by independently sampling from the learned categorical distribution at each autoregressive step [6]. Since most probability mass concentrates in the dominant direction, repeated samples are likely to cluster on the same route. The diversity penalty used in our diverse sampling variant explicitly suppresses this clustering by downweighting observations near already-chosen branches, forcing each sample to explore a different plausible route. The cost is worse *top-1* accuracy for the diverse mode (13.39 vs. 9.13 km), since branches are pushed away from the most likely path, but the result is gains in *top-16* performance.

While our diverse sampling method slightly underperforms TrAISformer on *top-1* ADE, our beam search method exceeds both in that metric. To compute *top-1* using TrAISformer, we implemented argmax decoding over its categorical distribution. At route junctions, this per-step argmax selects the highest-probability bin independently at each attribute (lat, lon, SOG, COG), which can produce positions between modes where no vessel would travel. Our beam search scores complete paths by cumulative log-density, naturally favoring trajectories that stay on shipping lanes throughout.

Limitations. Our method cannot predict rare maneuvers (e.g., a turn taken by <1% of historical traffic at that location) without trajectory history indicating intent. The diverse sampling mode's CRPS is worse than TrAISformer's due to the diversity penalty spreading samples; however, our beam search variant achieves comparable CRPS (8.44 vs. 8.23 km) while lacking the multi-modal coverage of the diverse mode. No single inference mode dominates all metrics simultaneously.

Practical implications. The method requires no GPU, no training loop, and no hyperparameter tuning beyond kernel bandwidths. For a new geographic region, deployment requires only accumulating AIS data: approximately two weeks of observations (≈14,000 transitions) suffices to match TrAISformer's accuracy trained on 11 months of data. The transition table can be updated incrementally as new data arrives without rebuilding.

Conclusion. We presented a training-free trajectory prediction method based on state-conditioned transition lookup with diversity-penalized sampling. The method matches or exceeds a 57M-parameter transformer on both deterministic and multi-modal prediction metrics while requiring no GPU, generalizing gracefully under data sparsity, and using zero learned parameters. The core insight is that multi-dimensional kernel conditioning (bearing, speed, temporal) combined with explicit diversity promotion transforms classical nearest-neighbor prediction into a competitive multi-modal predictor—without learning a single parameter. The method's nonparametric nature suggests applicability to other movement domains (urban mobility, aviation, wildlife tracking); we leave cross-domain evaluation to future work.

REFERENCES

[1] I. S. Abramson. On bandwidth variation in kernel estimates—a square root law. *The Annals of Statistics*, 10(4):1217–1223, 1982.

[2] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. IEEE/CVF CVPR*, 2018.

[3] S. Hexeberg, A. L. Flaten, B.-O. H. Eriksen, and E. F. Brekke. AIS-based vessel trajectory prediction. In *Proc. 20th Int. Conf. Information Fusion*, 2017.

[4] M. Liang, R. Gao, Y. Zhao, and T. Shi. Vessel trajectory prediction in maritime transportation: Current approaches and beyond. *IEEE Trans. Intelligent Transportation Systems*, 23(12):22527–22542, 2022.

[5] E. A. Nadaraya. On estimating regression. *Theory of Probability and Its Applications*, 9(1):141–142, 1964.

[6] D. Nguyen and R. Fablet. TrAISformer—a generative transformer for AIS trajectory prediction. *arXiv:2109.03958*, 2021.

[7] G. Pallotta, M. Vespe, and K. Bryan. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy*, 15(6):2218–2245, 2013.

[8] B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In *Proc. 11th Int. Conf. Information Fusion*, 2008.

[9] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. In *Proc. AAAI*, 32(1), 2018.

[10] G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964.

[11] Danish Maritime Authority. Historical AIS data. https://dma.dk/safety-at-sea/navigational-information/ais-data.

[12] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. 18th Int. Conf. World Wide Web*, pp. 791–800, 2009.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.