
Hurricane Tracking with the Kalman Filter

Diego Arevalo Fernandez, Charlie Ngo, Diego Osborn, Sardor Sobirov
University of California San Diego
GitHub Repository: Hurricane Paths Repo

1 Problem Description

This project addresses the problem of accurately forecasting the trajectories of hurricanes using probabilistic modeling. The importance of an accurate forecasting model is mostly in the context of natural disaster aid. As having an accurate forecasting model is *essential*, as the smallest errors may lead to people evacuating safely or *not*¹.

Due to complex atmospheric interactions and incomplete observational data, hurricane tracks are increasingly difficult to forecast given the inherent noisiness. Even the best datasets will still contain substantial noise and heterogeneity across hurricanes². A deterministic model will most likely also fail in this case, as errors in previous iterations would propagate further, thereby increasing the overall error of the prediction even more. As such, to combat the issues of uncertainty across observations and in a probabilistic manner, we want to follow in the footsteps of Gordon Bill³ and ask the question: **Can a Kalman filter-based probabilistic framework improve the estimation of hurricane position and uncertainty over time?**

2 Data Sourcing and Processing

2.1 Data Sourcing/Dataset Description

We sourced our data from IBTrACS (International Best Track Archive for Climate Stewardship) produced by the NOAA, or the National Centers for Environmental Information. This dataset contains a full record of tropical cyclones around the world from 1842 up until 2025⁴. This dataset contains 722,040 entries representing 13,530 unique storms, with each entry representing a single storm's characteristics (represented as columns such as seasonality, windspeed, latitude/longitude, etc.).

2.2 Data Loading

One of the main challenges we faced with this dataset is that it contains two headers: The first one contains measurement units for each column, and the second one contains column names. This provides useful metadata for the measurements and characteristics of each hurricane at a specific time, but it can introduce potential issues with the associations between the columns and their respective units. To address this, we passed the dataset through a function `load_ibtracs()`, which gathered column names and units. It also renamed column names by replacing all spaces with underscores and setting all characters to lowercase, as well as converting all dates to datetime. Another issue was that blank entries in the dataset were represented by blank strings instead of null values. Two categories of missing data were discovered: Certain storms had missing velocity entries, which is because those storms only had one observation/time block attached to them, and therefore, velocity cannot be computed with a lack of future observations. We decided to drop these entries, as these storms would be redundant with a lack of velocity and because they only add up to a total of 80 observations/storms out of more than 13000 observations. We assume that other missing values represent random non-existent or faulty measurements, which are not associated with a particular variable or category.

¹(DeMaria et al., 2022).

²(Kruk et al. 2010).

³(Bril, 1995).

⁴(IBTrACS, 2021).

2.3 Velocity Dynamics

The main preprocessing task focuses on the calculation of hurricane velocities and coordinate transformations. While storm speed and direction are already represented in the original dataset, we’re not able to verify the accuracy of these measurements and their uncertainties. Therefore, we estimated storm velocity based on the consecutive position observations of said storm. We accounted for Earth’s spherical shape through the use of haversine distance calculations that are able to handle longitude wrapping and latitude-dependent scaling of longitudinal distances. To improve stability and interpretability, we computed velocities for corresponding six-hour intervals instead of three-hour intervals.

2.4 Feature Engineering

A key assumption of the Kalman filter is that motion evolves according to linear dynamics in flat, Euclidean space⁵. However, latitude and longitude lie on Earth’s curved surface, so differences in degrees do not correspond to uniform physical distances. Because of this, we cannot directly treat changes in latitude or longitude as straight-line displacements within a single linear coordinate system. To address this, we take the first observation of a storm and treat it as the origin of a local flat coordinate system. For each subsequent observation, we use spherical geometry to compute the true distance traveled, and then decompose this displacement into its east–west (x) and north–south (y) components. Since our main goal is to accurately capture storm behavior and patterns unique to certain storms, we calculated the track curvature for each storm, which measures the rate at which the direction of a storm is changing. Therefore, we measured the angle between consecutive velocity vectors and clipped extreme values, which can distort the model and complicate scaling/computations. Additionally, the proximity of a particular storm to a large landmass can alter the storm’s behavior and lead to erratic patterns⁶. Therefore, we created a binary variable to track if storms were close to land or not, with a cutoff of 200 km. Subsequently, the direction of movement of a storm is noticeably influenced by its latitude. For example, storms in the tropics tend to move east to west along the trade wind belt, but once they move north or south, storms tend to curve upward/downward towards the nearest pole, and then fully reverse and move eastward, due to the trade winds/mid-latitude westerlies. It is also important to categorize storms based on their intensity (e.g., tropical storm vs hurricane). Since the stage at which a storm finds itself is another factor that influences motion patterns, we represent these characteristics as variables through the use of one-hot encoding.

2.5 Observation Characteristics

It is important to note that not all storms are neatly represented through continuous, evenly spaced time intervals. While the vast majority of storms in the dataset are continuous, there is a small minority of older storms that precede satellite imagery, and some of these storms were recorded in irregular time blocks. Because of this, we flagged these storms for later potential exclusion. Additionally, there is significant variance in the number of observations per storm, which can directly impact the predictability of a storm. There also seems to be a geographic imbalance of storm locations, with a large amount of observations coming from the Western Pacific Ocean (about 241,000), while the South Atlantic only contains 119. This suggests that region-specific modeling or weighting may be needed to capture distinct steering behaviors across ocean basins.

3 Modeling and Inference

As mentioned previously, to solve this problem of accurately forecasting the trajectories of hurricanes, we opted into using a Kalman filter, which is essentially a continuous-state Hidden Markov Model (HMM) designed for tracking systems that evolve smoothly over time⁷. The Kalman filter, in our case, works by having continuous vectors, representing a given hurricane’s position and velocity in kilometers, as our hidden states. Meaning, at each time step, the hidden state (where the hurricane truly is and how fast it’s moving) evolves according to a linear transition model with Gaussian noise, and we observe a noisy measurement of that state (the reported noise from the best-track data).

Similar to the forward algorithm in HMMs, the Kalman filter computes $\mathbb{P}(x_t|y_{1:t})$, the probability distribution over the current hurricane state given all position observations up to time t ⁸. The only

⁵(Bril, 1995).

⁶(Bril, 1995).

⁷(Roweis & Ghahramani, 1999).

⁸(Wolfe, 2011).

key difference is that because everything is linear and Gaussian, the Kalman filter can compute these probability distributions exactly using matrix operations rather than summing over discrete states.

Our probabilistic model represents each hurricane as a sequence of hidden states over time, where each state x_t is a four-dimensional vector:

$$\{x_{km}, y_{km}, vx_{km}, vy_{km}\}, \quad (1)$$

in which the first two components represent the storm’s east-west and north-south position relative to where the storm started, in km, and the last two components represent the storm’s velocity (how many km it’s moving east-west and north-south every six hours). This state evolves over time according to the Markov property, in which the next state, x_{t+1} , depends solely on the current state, x_t , and not on any earlier states. The model’s dependency structure follows that of the HMM. As we have a hidden state at time t that depends only on the hidden state at time $t - 1$ through the transition model, and the observation at time t depends only on the hidden state at time t through the observation model.

We implemented several variants of the Kalman filter for comparison. We had a baseline standard linear Kalman filter model that used constant process noise and observation noise parameters throughout all storms. Our improved model attempted to adapt the process noise, Q , based on storm features such as track curvature and land proximity, making it a “feature-adaptive” Kalman filter. This approach allowed us to establish a baseline (constant- Q) model and test whether adaptive enhancements actually provided meaningful improvements.

We can define the Kalman filter with two equations. The first of which is the state transition model (how the hidden state evolves from one time step, t , to the next):

$$x_{t+1} = A \times x_t + w_t, \quad (2)$$

in which x_t is our four-dimensional state vector [Equation 1] at time t , A is our transition matrix, and w_t is the process noise drawn from a Gaussian distribution with covariance Q : $\sim \mathcal{N}(0, Q)$. This equation captures the constant velocity dynamics, where the new position equals the old position plus velocity. The second of which is the observation model (how our measurements relate to the true hidden state):

$$y_t = H \times x_t + v_t, \quad (3)$$

in which y_t is our two-dimensional observation vector (the reported position in km), H represents the observation matrix, which is just the position components from the full state, and v_t is the observation noise drawn from a Gaussian distribution with covariance R : $\sim \mathcal{N}(0, R)$. This equation essentially states that what we observe is the true position “corrupted” by measurement noise.

Our model’s parameters are:

- The transition matrix A , which is a 4×4 matrix that implements the constant velocity motion, such that position updates by adding velocity, while velocity remains constant between time step. This matrix is not actually learned from data but is instead derived directly from the physics of motion, where distance = velocity \times time.
- The observation matrix H , which is a 2×4 matrix that selects *only* the position components (x and y coordinates) from the full state vector, thereby ignoring the velocity components. This is because we *only* observe storm positions (not velocities directly), meaning the matrix is fixed based on this measurement structure rather than learned from data.
- The process noise covariance Q , which is a 4×4 matrix that represents how much the state can randomly deviate from the constant velocity model at each time step, is estimated from the training data by computing the covariance of the prediction errors. Q has large values, meaning that the storms would frequently accelerate, decelerate, or turn in ways that violate the constant velocity assumption. Our adaptive version model scaled Q by factors between $1.0\times$ and $3.0\times$ based on storm features as an attempt to increase uncertainty when storms would turn sharply or approach land.
- The observation noise covariance R , which is a 2×2 matrix that reflects the inherent uncertainty in the best-track data (citation), is estimated at approximately 265 km² variance for east-west position and 169 km² variance for north-south position.

The inference algorithm that we implemented was the exact inference algorithm using the Kalman filter equations, which is the continuous-state equivalent of the forward algorithm (citation). Through

this inference algorithm, the model would compute the exact posterior distribution $\mathbb{P}(x_t|y_{1:t})$ at each time step. And because the model is linear with Gaussian noise, this posterior will *always* be Gaussian as well⁹. The algorithm itself had two steps at each time step:

- The predict step, which propagated the previous state forward using the transition model, thereby computing $\mathbb{P}(x_t, y_{1:t-1})$, the one-step-ahead predictive distribution that served as the forecast,
- and the update step, which incorporated the new observation y_t to refine the state estimate by computing the prediction errors, then using the Kalman gain¹⁰¹¹ to update both the mean state estimate and the covariance. Resulting in producing $\mathbb{P}(x_t|y_{1:t})$, the filtered distribution.

For learning, we estimated the parameters Q and R from the training data rather than incorporating domain knowledge and manually setting them ourselves. To do this, we ran the Kalman filter on roughly 200 storms from the training set using initial guesses, recorded the prediction residuals at each step, and calculated their sample covariance. This empirical covariance then served as our final estimate for Q and R. Given that we were finding the Q and R that best explained the observed prediction errors across the training set, we can assume this step to be a form of maximum likelihood estimation (MLE). We also performed a form of hyperparameter tuning for the adaptive-Q scaling factors by using a validation-set approach, in which we tried many scaling parameter combinations, evaluated forecast accuracy on held-out validation storms for each combination, and selected the parameters that minimized forecast error.

The problem that we are trying to solve is inherently filtering and forecasting. Filtering refers to computing $\mathbb{P}(x_t|y_{1:t})$, which is, as mentioned, what the forward algorithm does for HMMs. And forecasting refers to computing $\mathbb{P}(x_{t+k}|y_{1:t})$, which is predicting future states given the current observations. For model evaluation, we generated one-step-ahead forecasts ($k = 1$) by predicting x_{t+1} from x_t , then comparing this predicted position to the actual observed value at $t + 1$ to measure forecast error. For multi-step forecasts, we iteratively applied the transition model without updating with new observations, which inherently caused uncertainty to grow as Q noise would accumulate at each time step.

4 Results and Discussion

4.1 Forecast Performance

We evaluated the Kalman filter using sliding-origin methodology¹². on 20 test storms (555 forecast instances). Table 1 summarizes forecast accuracy across lead time marks.

Table 1: Track Forecast Error Summary Statistics

Lead Time	Mean Error (km)	RMSE (km)
6 hours	11.90	15.86
12 hours	24.53	34.50
24 hours	58.13	83.79
48 hours	158.95	216.74
72 hours	286.08	379.35

The results demonstrate short-term accuracy with errors approximately doubling every 12 hours, showing quadratic growth from accumulated growth in noise. The constant velocity-model velocity-model¹³. was able to reproduce the persistent motion of the storms effectively within the 6-24 hour window, deteriorating rapidly post the 48 hour mark, due to recurvature and change in steering flow (site). Filtering evaluation (continuous observation updates) on 2,680 test storms yielded mean error of 21.13 km and RMSE of 26.85 km, establishing the upper bounds of achievable performance.

⁹It can be represented exactly by just a mean vector (best estimate of the state) and a covariance matrix (uncertainty about that estimate).

¹⁰A matrix that optimally weights how much to trust the new observation versus the prediction.

¹¹(Franklin, 2020).

¹²(Cangialosi & Franklin, 2014).

¹³(Roweis & Ghahramani, 1999).

The median error of 16.85 km falls below the mean as shown in Figure 1, indicating right-skewed distributions inferring that difficult cases produce large errors (maximum 203 km).

4.2 Error Distribution Analysis

A detailed examination of forecast error distributions reveals substantial deviations from the Gaussian assumptions underlying Kalman filter theory,¹⁴. Across all time marks, the empirical densities exhibit sharp peaks near zero and long, heavy right tails as show in histograms in Figure 2. Kernel density estimates diverge noticeably from the fitted normal curves, with the empirical KDE showing a much sharper peak near zero error and a slower decay along the x-axis. Meaning the Gaussian distribution consistently overestimates moderate errors, while underestimating the closer it gets to zero error. Similar to the finding in Table 1 these discrepancies grow with each passing time mark, implying the forecast errors are fundamentally non-Gaussian. As a result, the covariance matrices produced by the Kalman filter underestimate tail risk¹⁵, and Gaussian-based confidence ellipses would yield overconfident forecast cones that fail to represent the true spread of storm trajectories.

4.3 Residual Analysis and Model Validation

Despite the forecast errors, the innovation scatter plot ($v_t = y_t - H\hat{x}_t|t-1$) Figure 3 shows that the residuals remain centered near zero, symmetrically spread. Indicating no strong systemic bias in a direction, confirming the constant-velocity transition(site) model does not possess drift¹⁶.

Table 2: Innovation Ratio Diagnostics and Predictive Power

Feature	High/Low Innovation ² Ratio	Predictive Power
Track Curvature	~ 1.0	Weak
Land Proximity	~ 1.0	Weak
Motion Regime	~ 1.0	Weak
Latitude Regime	~ 1.0	Weak

However, innovation analysis reveals that our adaptive features (track curvature, land proximity, motion regime, latitude regime) show weak correlation with forecast uncertainty. The ratio of innovation squared between high-feature and low-feature periods approximates 1.0 across all features as seen in Table 2 below, and regression models predicting innovation variance achieve $R^2 < 0.10$.

This explains why adaptive-Q produces limited improvement; these features don't actually identify when the filter makes large errors, as seen in Table 3 below.

4.4 Baseline vs. Adaptive-Q Comparison

Table 3: Baseline vs Adaptive-Q RMSE Across Lead Times

Lead Time	Baseline RMSE (km)	Adaptive-Q RMSE (km)	Improvement
6 hours	15.86	15.58	1.8%
12 hours	34.50	34.05	1.3%
24 hours	83.79	83.04	0.9%
48 hours	216.74	216.74	0.0%
72 hours	379.35	379.50	-0.04%

The adaptive-Q model dynamically adjusts process noise according to storm characteristics¹⁷. but produces only marginal performance gains. The difference between Baseline RMSE and Adaptive-Q RMSE for every time mark was below one kilometer.

The RMSE curves Figure 4 for both models are nearly indistinguishable at all lead times, with the adaptive version producing only marginal improvements before converging entirely with the

¹⁴(Bril, 1995; Roweis & Ghahramani, 1999).

¹⁵(Wolfe, 2011).

¹⁶(Franklin, 2020).

¹⁷(DeMaria et al., 2022).

baseline by forty-eight hours. The scatter plot of adaptive versus baseline forecast errors confirms this lack of differentiation, nearly every point lies directly along the diagonal line, indicating that individual forecasts are effectively unchanged regardless of whether Q is held constant or modulated by curvature, land proximity, or motion regime. Corroborate the earlier innovation analysis

4.5 Storm Characteristics and Forecast Accuracy

Forecast accuracy varies strongly with storm length¹⁸. Storms with fewer than ten observations produce highly unstable predictions, reflecting insufficient temporal context for the filter to establish reliable estimates. In contrast, storms with long observational histories achieve significantly lower RMSE, and tightly concentrated distributions. The Figure 5 highlights this convergence: as the filter incorporates more observations, posterior covariances shrink and estimates stabilize. The distribution of mean errors across storms confirms that the filter remains largely unbiased¹⁹., with no directional drift.

4.6 Computational Performance

The adaptive- Q pipeline required approximately five to six hours of computation due to per-step feature extraction. This large computational overhead, combined with minimal performance improvement, concludes the conclusion that the primary limitation is not computational nor data-related but inherent in the limited predictive power of the selected adaptive features.

5 Conclusion

5.1 Limitations

Hurricanes are very complex systems influenced by *many* interacting factors that change constantly²⁰. And our model assumes that hurricanes will continue moving in roughly the *same* predictable patterns (same direction at same speed). This works reasonably well for short periods (6-12 hours), however, hurricanes don't behave like this for long²¹. Because hurricane motion becomes *genuinely* unpredictable over time, this makes our process that much more limited.

The Kalman filter we used is fundamentally *linear*, meaning we're assuming that the relationships between variables are also linear. However, hurricane motion is deeply *nonlinear*, so when a storm curves northward (recurvature)²², it is not following a "gentle" arc that can be *described* by our "simple" equations. The nonlinearity from hurricane paths can most likely be attributed to complex interactions with outside factors (e.g., jet stream, changes in surrounding pressure systems, shifts in steering winds, etc.)²³. Our feature validation analysis revealed that the characteristics we came up to account for the uncertainty in storm path when close to land or when the track is more "curved," don't *actually* correlate well with the forecast errors. Implying that the underlying reasons for uncertainty in the tracks are more *complex* than the features that we did come up with, and linear Kalman filters simply can't capture these nonlinear dynamics as well.

Even though we had over 700,000 observations from 13,530 storms, there are inherent limitations with the IBTrACS "best track" data as the "best track" estimates represent post-analysis estimates of where the storms *actually* were, meaning they're not *perfect* measurements²⁴. The observations also only capture the position of the hurricane every six hours, meaning during those time gaps, it is *entirely* possible that the storms can undergo significant changes with respect to their trajectory that the model would never be able to see directly. Our data also uses historical data that spans nearly two centuries, meaning that because the measurement quality was much worse in those earlier records, when satellite imagery did not exist, the data quality adds another layer of uncertainty that makes tracking even harder to account for.

Every forecast step made was built on the previous one, meaning errors are bound to compound over time. If our model made a small error in the first prediction step, the next prediction would start from

¹⁸(Kruk, Knapp, & Levinson, 2010).

¹⁹(Roweis & Ghahramani, 1999).

²⁰(Martinez-Amaya et al., 2023).

²¹At 6 hours, our predictions were off by about 12 km on average, whereas by 72 hours, they're off by about 300 km.

²²(Hurricane FAQ – NOAA's Atlantic Oceanographic and Meteorological Laboratory, 2023).

²³(Barbero et al., 2024).

²⁴(International Best Track Archive for Climate Stewardship (IBTrACS) Technical Documentation 1, n.d.)

this slightly wrong position, leading to an even larger error. Meaning, by about 48-72 hours, so much uncertainty has accumulated, the predictions essentially become *unreliable*.

Because our dataset included storms from across the globe, each one is bound to come with their own unique characteristics. Some storms might move fast and straight, others slower, and others recurve sharply. Our model tries to handle this heterogeneity with a single set of equations and parameters. While we tried to adapt parameters based on storm regime, latitude, and others, there seems to be *too* much variability across storms for one model to capture effectively.

The important limitation to consider is that hurricane track forecasting has inherent predictability limits that no model can *truly* overcome without fundamentally different information. Our actual model is based *only* on past positions and velocity, but hurricanes are steered by atmospheric conditions that evolve independently of the hurricane itself²⁵. Meaning, to truly predict where a hurricane will go in days in the future, we'd most likely need to account for the *entire* atmospheric state across a large area, which we just don't have access to, to that extent.

5.2 Potential Improvements

In future iterations:

- We would like to pursue a synthetic data approach, which would have involved generating artificial hurricane tracks with known parameters in order to validate our model's ability to recover those parameters²⁶. Although, our extensive real-data evaluation across all of the storms still provided strong-enough empirical validation of model performance.
- If we were to move beyond Kalman filtering entirely, we would like to explore using Gaussian Process regression for fully probabilistic nonparametric forecasting.
- Beyond point forecasts, we would like to better characterize uncertainty and explore using ensemble forecasting with multiple initializations²⁷. We would also explore developing calibrated prediction intervals that adjust based on storm characteristics, such that the intervals would be narrower for predictable storms and wider for more erratic ones.
- To account for the data quality, we would like to explore training separate models for different time periods (pre-satellite era vs. modern era) or weighting more recent, higher-quality observations more heavily during parameter estimation.
- Addressing the 6-hour gap limitation, we would like to interpolate in-between hourly states using spline fitting or other smoothing techniques to create denser observations. Albeit, this approach would introduce added uncertainty that would need to be carefully modeled and accounted for.

6 Reflections & Contributions

For future students trying a similar project, we recommend thoroughly investigating your problem domain to build genuine curiosity and develop informed insights that will guide your project in meaningful directions.

I am Diego Arevalo Fernandez and I worked on the data preprocessing and feature engineering. I am Charlie Ngo and I worked on researching the scope of the project and what tools could be used. I am Diego Osborn and I worked on helping create the framework of the modeling and created graphs. I am Sardor Sobirov and I worked on the actual modeling of the Kalman filter.

We used Gen AI for coding help and debugging mostly, and we verified its outputs by observing if the advice worked or not. We also used it for research purposes with respect to possible applications in hurricane forecasting, and we verified its outputs by asking it to provide its sources for its recommendations and validating those aforementioned sources.

²⁵(Hurricane FAQ – NOAA's Atlantic Oceanographic and Meteorological Laboratory, 2023).

²⁶(Bril, 1995).

²⁷(Hurricanes: Science and Society: Ensemble or Consensus Models, 2020).

References

1. Barbero, T. W., Bell, M. M., Chen, J., & Klotzbach, P. J. (2024). A Potential Vorticity Diagnosis of Tropical Cyclone Track Forecast Errors. *Journal of Advances in Modeling Earth Systems*, 16(3). <https://doi.org/10.1029/2023ms004008>
2. Bril, G. (1995). Forecasting hurricane tracks using the Kalman filter. *Environmetrics*, 6(1), 7–16. <https://doi.org/10.1002/env.3170060103>
3. Cangialosi, J. P., & Franklin, J. L. (2014). *National Hurricane Center Forecast Verification Report*. NOAA/NHC. <https://www.nhc.noaa.gov/verification/>
4. Chan, J. C. L., & Gray, W. M. (1982). *Tropical Cyclone Movement and Surrounding Flow Relationships*. *Monthly Weather Review*, 110(10), 1354–1374. [https://doi.org/10.1175/1520-0493\(1982\)110<1354:TCMASF>2.0.CO;2](https://doi.org/10.1175/1520-0493(1982)110<1354:TCMASF>2.0.CO;2)
5. DeMaria, M., Franklin, J. L., Zelinsky, R., Zelinsky, D. A., Onderlinde, M. J., Knaff, J. A., Stevenson, S. N., Kaplan, J., Musgrave, K. D., Chirokova, G., & Sampson, C. R. (2022). The National Hurricane Center Tropical Cyclone Model Guidance Suite. *Weather and Forecasting*. <https://doi.org/10.1175/waf-d-22-0039.1>
6. Franklin, W. (2020, December 31). Kalman Filter Explained Simply. The Kalman Filter. <https://thekalmanfilter.com/kalman-filter-explained-simply/>
7. Hurricane FAQ – NOAA’s Atlantic Oceanographic and Meteorological Laboratory. (2023, June 1). Atlantic Oceanographic & Meteorological Laboratory. <https://www.aoml.noaa.gov/hrd-faq/>
8. Hurricanes: Science and Society: Ensemble or Consensus Models. *Hurricanescience.org*. <https://www.hurricanescience.org/science/forecast/models/modeltypes/ensemble/index.html>
9. International Best Track Archive for Climate Stewardship (IBTrACS). (2021, June 17). National Centers for Environmental Information (NCEI). <https://www.ncei.noaa.gov/products/international-best-track-archive>
10. International Best Track Archive for Climate Stewardship (IBTrACS) Technical Documentation 1. (n.d.). https://www.ncei.noaa.gov/sites/default/files/2021-07/IBTrACS_version4_Technical_Details.pdf
11. Kruk, M. C., Knapp, K. R., & Levinson, D. H. (2010). A Technique for Combining Global Tropical Cyclone Best Track Data. *Journal of Atmospheric and Oceanic Technology*, 27(4), 680–692. <https://doi.org/10.1175/2009jtecha1267.1>
12. Martinez-Amaya, J., Longépé, N., Nieves, V., & Muñoz-Marí, J. (2023). Improved forecasting of extreme hurricane events by integrating spatio-temporal CNN-RF learning of tropical cyclone characteristics. *Frontiers in Earth Science*, 11. <https://doi.org/10.3389/feart.2023.1223154>
13. Roweis, S., & Ghahramani, Z. (1999). A Unifying Review of Linear Gaussian Models. *Neural Computation*, 11(2), 305–345. <https://doi.org/10.1162/089976699300016674>
14. Wolfe, J. (2011). BP on Gaussian hidden Markov models: Kalman filtering and smoothing [Lecture notes]. Massachusetts Institute of Technology: MIT OpenCourseWare. https://ocw.mit.edu/courses/6-438-algorithms-for-inference-fall-2014/6a29d3d38474e9c62b746fa646899552_MIT6_438F14_Lec13.pdf

Appendix

Appendix 1: Graphs

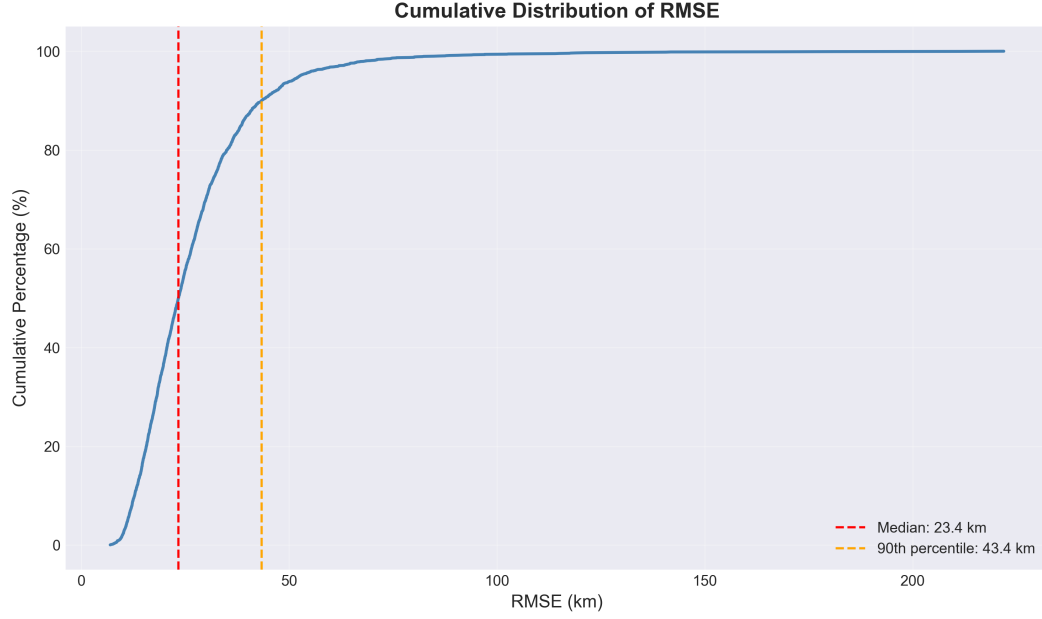


Figure 1: Cumulative distribution of track forecast RMSE across all forecast instances. Vertical lines mark the median, 75th percentile, and maximum error, summarizing the tail behavior of the baseline Kalman filter.

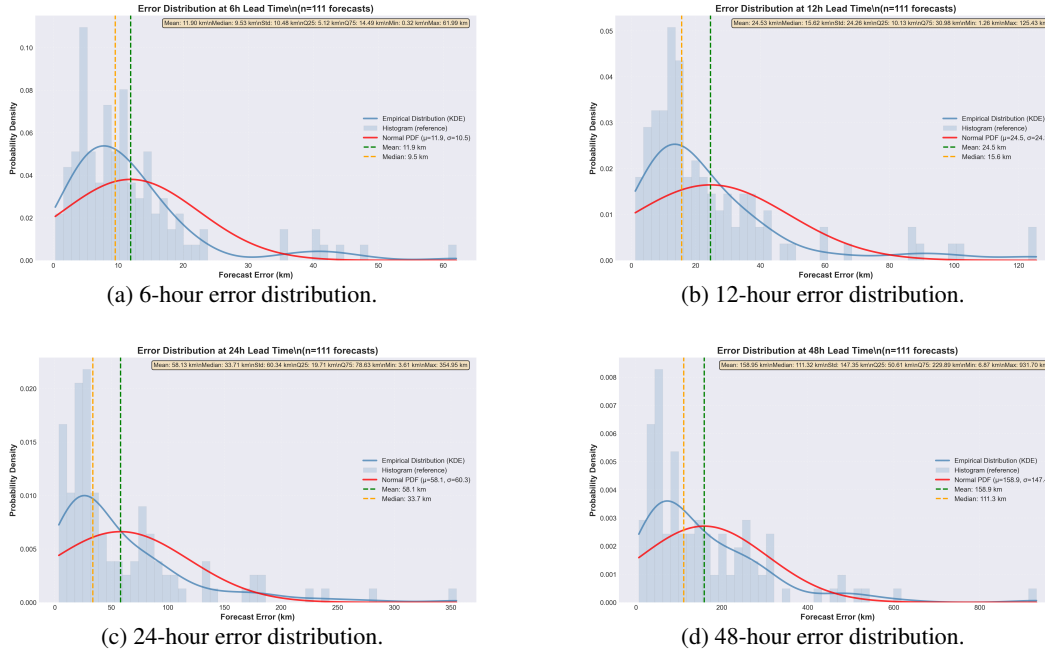


Figure 2: Forecast error distributions at four lead times (6, 12, 24, and 48 hours). Histograms, kernel density estimates, and fitted Gaussian curves highlight the increasing skewness and heavy right tails with longer lead time.

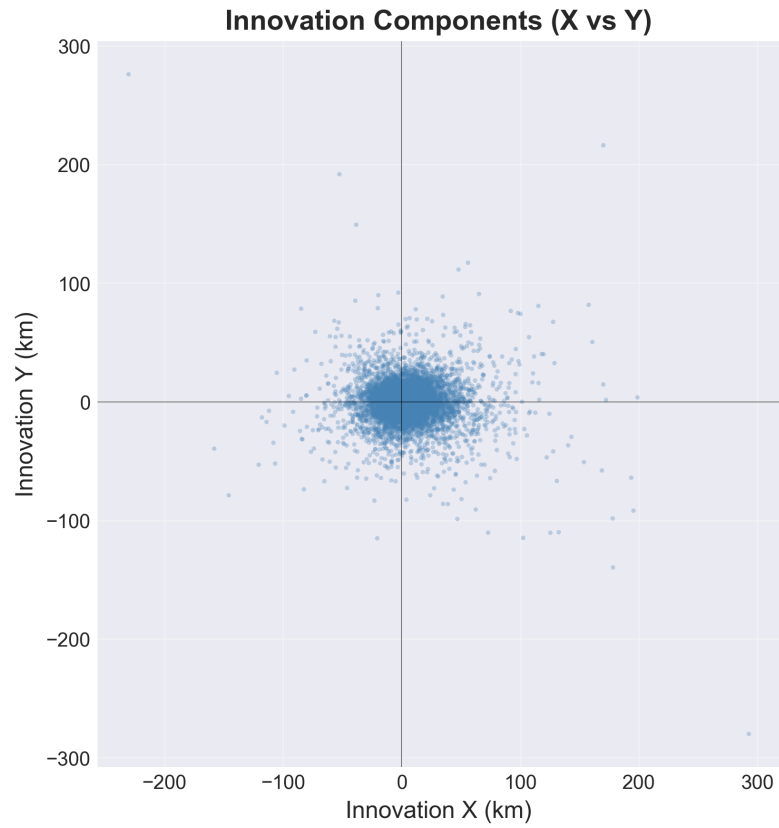
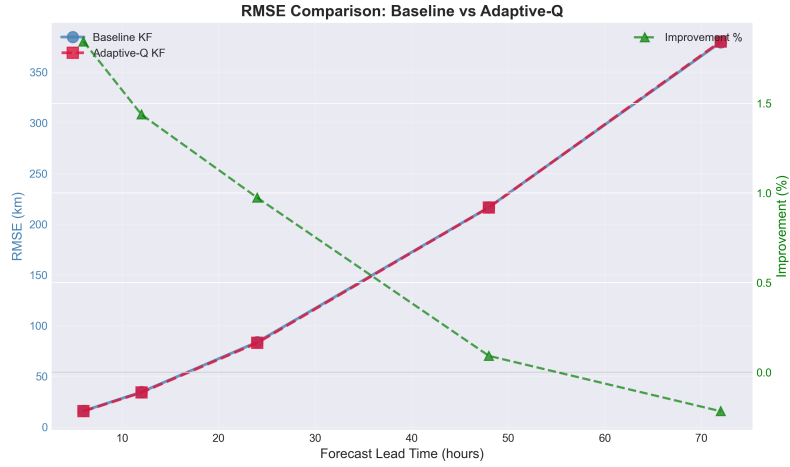
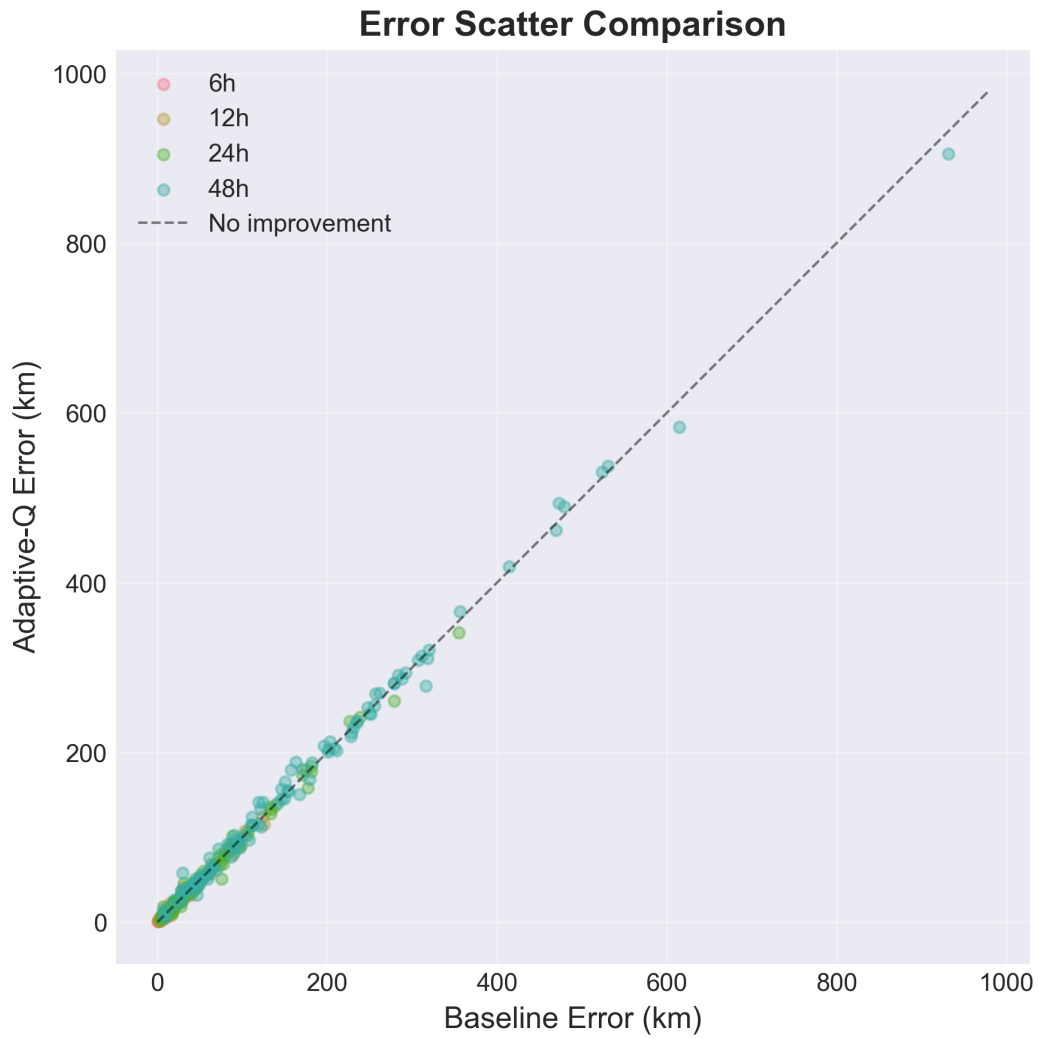


Figure 3: Scatter of innovation components in the along-track (X) and cross-track (Y) directions for all analysis times. The roughly circular, centered cloud indicates unbiased but heavy-tailed innovations.



(a) Lead-time RMSE comparison between baseline and Adaptive-Q filters.



(b) Error scatter comparison for baseline vs Adaptive-Q forecasts.

Figure 4: Comparison of baseline and Adaptive-Q Kalman filters. Panel (a) shows RMSE by lead time; panel (b) compares pointwise forecast errors, highlighting regimes where Adaptive-Q reduces large outliers.

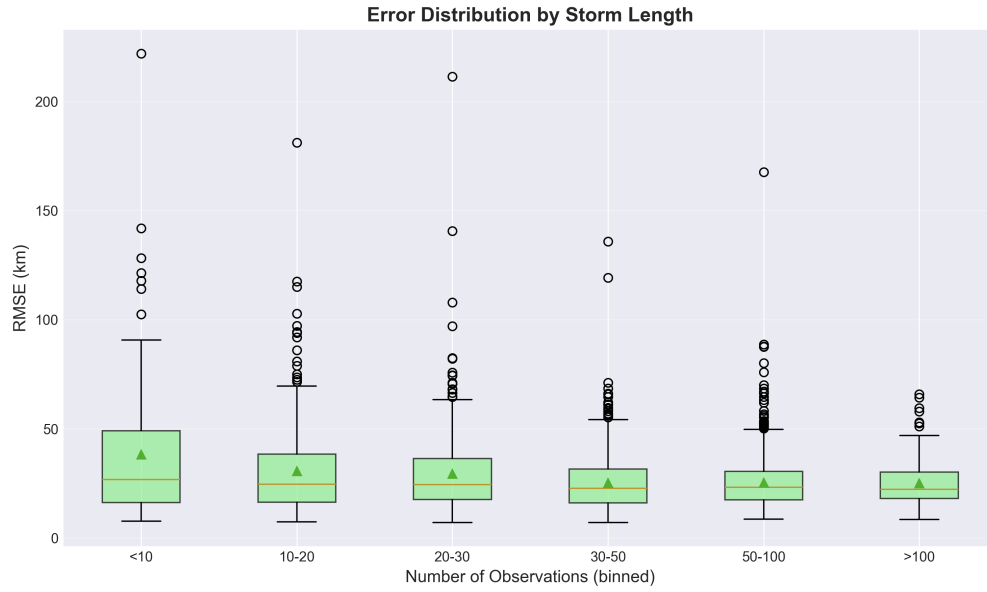


Figure 5: Relationship between track forecast error and storm length. The plot shows how storms with short observational histories tend to exhibit higher forecast error, reflecting limited information for estimating initial motion and dynamical regime. Longer storms generally show reduced error variance as the filter accumulates more observations and stabilizes state estimates.