https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

Prometheus: An On-line Tracker for Tropical Cyclones in ICON

Thibault Pierre Meier[1], Sylvaine Ferrachat[1], Iolène Bouzdine[1], Andrina Caratsch[1], Bernhard Enz[1], and Ulrike Lohmann[1]

[1]Institute for Atmosphere and Climate Science, ETH Zürich, Universitätstrasse 16, 8092 Zurich, Switzerland

Correspondence: Thibault Pierre Meier (thibault.meier@env.ethz.ch) and Sylvaine Ferrachat (sylvaine.ferrachat@env.ethz.ch)

Abstract.

Tropical cyclones (TCs) are among the most destructive weather events. To better understand them, numerical weather models are used to simulate those systems, which are then detected by designated tracking algorithms. This is generally done in a two-step process, where the tracking process is done in an off-line manner, after the weather simulation has terminated. This requires to save diagnostics data over the whole simulated domain, which are then processed by the tracking program and then by a processing tool to extract diagnostics quantities around the tracks. This presents the inconvenience to require massive amounts of intermediate data, which can rapidly become prohibitively large, with increasing spatial and temporal resolution, especially in the context of performing ensemble simulations. In this paper, we present an innovative solution to this problem, by implementing an on-line version of the Enz et al. (2023) tracking algorithm: the TC detection is performed directly within the numerical weather prediction and climate model ICON (Zängl et al., 2015) and any diagnosed quantity from the ICON model can be output in a user-defined region around each detected TC centre. We conduct a detailed comparison analysis between the off-line and the on-line tracking algorithm versions. Besides the already exposed gain in storage footprint, we show that the on-line version allows a higher track detection calling frequency, yielding more accurate results.

1 Introduction

Tropical Cyclones (TCs) that make landfall are among the most devastating natural disasters around the world, with millions in damages and thousands of deaths per year in the United States alone (NOAA, 2024). Numerical weather and climate models offer many ways to help improve our knowledge about TCs. The models can be used to better understand the structure of TCs using free running simulations or to improve understanding of past TCs using climate reanalysis (Bourdin et al., 2026).

These numerical models however, are usually set up to cover an entire oceanic basin – such as the Atlantic – or run globally and trying to manually spot TCs in the simulation's output data quickly becomes cumbersome. In order to facilitate the analysis of TCs using weather and climate models – such as ICON (Zängl et al., 2015) – many different algorithms were designed with the goal of detecting and tracking the trajectories of TCs in the model output.

The basis for those algorithms is derived from the prerequisites for TC formation as well as the features of TCs. This includes criteria for minimum sea-surface temperature at genesis, minimum vertical vorticity at the centre, a warm temperature anomaly in the core of the TC, low wind-shear, high relative humidity and a local low pressure anomaly (Bourdin et al., 2022). One of

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

the main differentiating factors among those algorithms is in how they define the storm centre. While some use the pressure minimum as indicator for the TC centre (Ullrich et al., 2021), others use the vorticity maximum (Hodges, 1994). Yet others use their own developed metric (Tory et al., 2013). The unifying factor in all tracking schemes is their sensitivity to changes in the threshold values (Enz et al., 2023; Zarzycki and Ullrich, 2017). One avenue to avoid that is to use deep learning algorithms that are trained on reanalysis data and have no physical thresholds to be defined (Galea et al., 2023). The other avenue is to employ a scheme that uses multiple threshold values in order to capture a more comprehensive set of tracks (Camargo and Zebiak, 2002).

The tracks are then generated by stitching together the detected points, which is usually done by using a closeness criterion (Bourdin et al., 2022). A more flexible criterion, based on smoothness optimisation was defined by Hodges (1999). More detailed studies and comparisons of the different tracking algorithms can be found in Bourdin et al. (2022), in the appendix of Ullrich and Zarzycki (2017) and for the deep learning algorithm in Galea et al. (2024). In this paper, we use the tracking algorithm introduced by Enz et al. (2023) – called Epimetheus – as a basis for our on-line implementation – called Prometheus – which is discussed in subsection 2.1.

The current (off-line) implementation of Epimetheus can be described as a three-phase process: 1) running the numerical model and generating model output over the entire simulation domain, 2) reading model output into the tracking program and generating TC tracks, 3) using the track data to extract diagnosed model data around each track. This approach has the drawback of a large data overhead, as one needs to output data on the entire simulation domain while most of it is void of TCs. With grid resolutions at km-scale the data overhead becomes even more pronounced. The on-line tracking approach is one solution to the data problem, with other approaches including running a first simulation on a coarser grid and then generating a smaller high resolution grid around the TC track for a second run (Senf and Cremer, 2026). Both approaches have unique additional advantages outside of the reduced storage burden – such as computational savings from Senf and Cremer (2026) and track improvements from Prometheus – meaning they can also be used complementarily.

The main goals of this paper are motivated by the data overhead of the off-line implementation and can be summarised as follows:

1. Integrate the algorithm into the ICON time integration loop (take it on-line). This is discussed in section 2.2.
2. Use the tracking information to restrict the output regions to capture only the entire TC and the necessary environmental parameters to reduce output file sizes (section 2.3).
3. Compare the off-line (Epimetheus) and the on-line (Prometheus) implementations and discuss the differences (section 4).
4. Analyse new possibilities offered by the on-line nature of Prometheus (section 5).

A comparison of the respective flowcharts of the Epimetheus and Prometheus algorithms is depicted in Figure 1.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

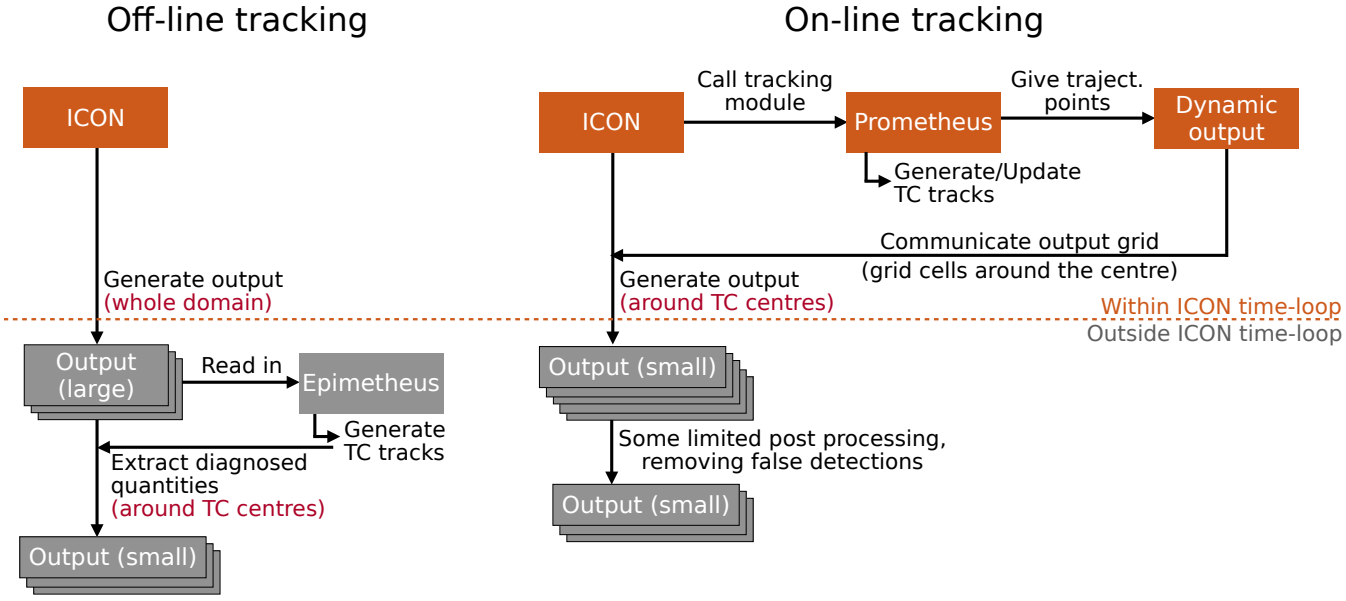


Figure 1. Flowchart visualizing the differences in operation of the algorithms. Orange are all parts that are run inside the ICON simulation, grey are outputs/phases that occur outside of ICON.

2 Method

To achieve the goals of taking the TC trackings algorithm on-line and to dynamically restrict the output region(s), we need to implement two modules that connect directly to the ICON time-loop. The first is the tracking module Prometheus, which is the on-line implementation of the Enz et al. (2023) Epimetheus algorithm. The second module restricts the output region to only capture the area around the tracked TCs.

2.1 Off-line tracking (Epimetheus)

The Enz et al. (2023) algorithm uses three tracking criteria to check whether a grid-cell is a TC centre:

1. Pressure criterion: "The sea level pressure must exhibit [one] local minimum within a given distance ($p_{s,dis}$)"
2. Vorticity criterion: "The vertical component of the relative vorticity must exceed a threshold value within the lower troposphere (ξ_{min})"
3. Warm core (temperature criterion): "The $300\,\mathrm{hPa}$ temperature directly above the sea level pressure minimum must exceed the mean $300\,\mathrm{hPa}$ temperature within a given distance (T_{dis}) by a certain value (ΔT_{core})"

The first criterion is used to determine the centre of the TC. The remaining criteria are checked sequentially in the order listed, continuously disqualifying low pressure centres not belonging to TCs. In a second step, the resulting TC centres at different

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

Variable		Threshold Values				
$p_{s,dis}$	[km]	50	100	150		
ζ_{min}	[s^{-1}]	10^{-6}	10^{-5}			
ΔT_{core}	[K]	0.5	0.75	1	1.25	1.5
T_{dis}	[km]	50	100	200	300	400

Table 1. Threshold parameters used for tropical cyclone tracking (Enz et al., 2023)

time-steps are stitched together by connecting the closest TC from the previous time-step. To ensure the centres belong to the same TC, a maximum radius of movement is allowed corresponding to a translational velocity of $20\,\mathrm{m\,s^{-1}}$ over 6-hourly time-steps. Those two steps result in a set of TC tracks. One of the novelties of the algorithm is that it uses multiple threshold values for its criteria (see Table 1). The reasoning behind this is that the strict criteria disqualify false positives and the loose criteria capture the less pronounced genesis and decay of a TC. Applying the two steps for each of the threshold combinations results in 150 sets of tracks. These are then merged in a third step, where the two last constraints are applied:

4. A minimum lifetime criterion of $18\,\mathrm{h}$ ensures that short lived systems are not tracked as TCs.
5. A minimum detection criterion requires a track to be detected by at least $10\,\%$, $20\,\%$ or $50\,\%$ of threshold combinations for tropical depressions, tropical storms and hurricanes, respectively (Enz et al., 2023).

2.2 On-line tracking (Prometheus)

There are four main differences (besides converting the code from Python to Fortran) that impact the implementation and the application of the algorithm, when porting the algorithm directly into ICON as a set of submodules in the source code:

1. Grid: ICON uses an unstructured triangular grid as opposed to the regular grid used for Epimetheus (Zängl et al., 2015).
2. Decomposed domain: to facilitate parallelism, ICON decomposes the simulation domain into different parts, each owned by one process (PE) (Prill et al., 2023, pp.190).
3. Temporal information: Prometheus only has access to data from the current simulation time step. Past data needs to be specifically preserved, and future data is unavailable.
4. Time steps: Epimetheus has access to ICON output generated at 6-hourly intervals (simulation time), while the Prometheus has access to data at every simulation time step.

While the first three differences provide challenges for implementation, the last one opens up the possibility of improved tracking accuracy, by calling the algorithm more frequently. These changes also lead to a slightly altered procedure shown in Figure 2.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

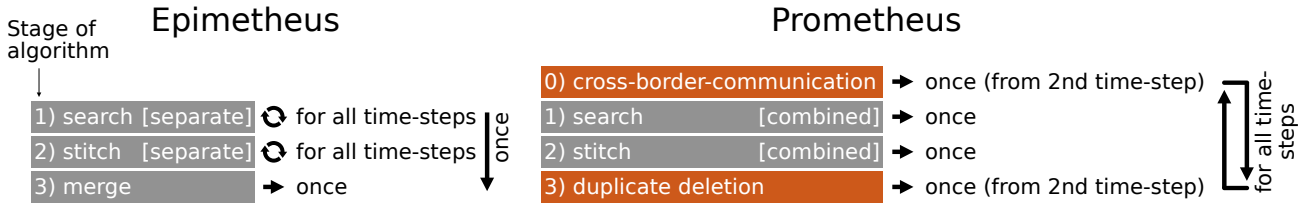


Figure 2. Overview of the stages in the tracking algorithms. Left: Epimetheus algorithm computing the first two stages on each threshold combination separately. Right: Prometheus algorithm where those stages are done on the combined threshold values. The new stages (orange) are required because of the domain decomposition. Merge in Epimetheus is needed because the first two stages are done separately on each threshold combination.

Adapting the algorithm to the triangular grid structure mainly affects the detection step, as cells in a certain radius around a potential TC centre need to be checked (for example, the pressure minimum: the algorithm checks that the pressure at the centre is lower than in any cell within $50\,\mathrm{km}$ to $150\,\mathrm{km}$). With the regular grid, cell indices around the centre can be deduced, while in the native grid we only have access to the neighbouring cells. To adapt to this, a recursive depth-first-search (DFS) algorithm is deployed, traversing all grid cells within the given radius around the center.

The decomposed domain generates two difficulties: for checking criteria – since cells that need to be checked can be in other domain parts, and for stitching TC tracks – since TCs can move across domain parts. In case of the detection step, the DFS algorithm is deployed on all domain parts that are reached by the search radius around the centre. The message passing interface (MPI) is used to collect the necessary information on the home-PE (domain part where the centre is located), where the exceedance of the thresholds is ultimately evaluated.

To keep most of that computation local to the PE, tracked TCs are stored in a linked-list on the PE where the TC is currently located. This means the data-structure containing the TC needs to be moved to a different PE, should the TC move to another domain part. As the next location of the TC cannot be predicted, the only available metric is the maximum translation speed and the corresponding "disc" of possible next locations. To cover all possible domain parts, a copy of the TC data-structure is sent to all PEs whose domain parts are close enough to the TC. After the next stitching event it is determined on which PE domain part the TC actually moves or remains. All the duplicate TC data-structures that were generated are subsequently deleted. Figure 3 showcases this two-part process. The first part consists of the communicating stage, which is done before the stitching stage using the TC locations from the previous tracking step. Part two consists of the duplicate deletion stage and is done after the stitching stage with the updated TC locations.

The third difference affects the detection criterion, which Epimetheus checks along the entire track at the end after all tracks have been stitched together. Using Prometheus, this provides a dilemma: either output all tracks, even if minimum lifetime and minimum detection is not yet reached and delete false positives a posteriori using a python script, or only start outputting

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

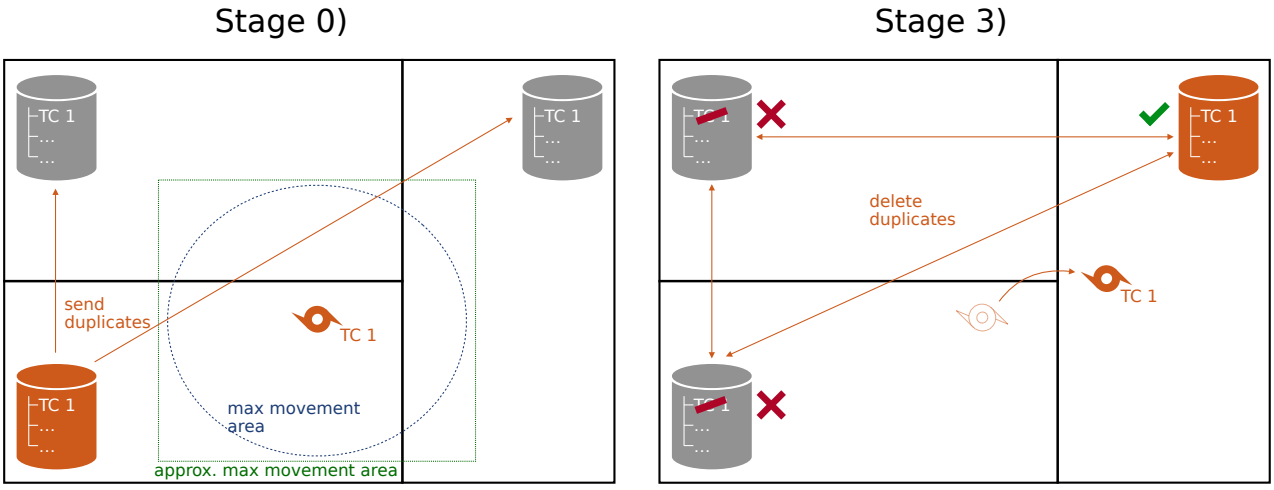


Figure 3. Process of cross-border-communication that is needed in case a TC moves between domain parts. Left: intersections of approximate maximum movement area (green dotted line) and other (non-home) domain parts are checked and a copy of the TC is sent to the PEs that belong to the respective domain parts. Right: after the next position is known. Communication between all PEs with a copy of the TC determines the new owner. The duplicates are deleted. Both stages happen during the same tracking step.

tracks after a track also fulfils the fourth and fifth criteria. Because TC genesis may be lost in the latter approach, the former is chosen.

Lastly the effect of having access to fields updated at each simulation step, is that the frequency at which the tracking algorithm is called can be made as high as each simulation-step. There is a practical limit to how often it should be called because the TCs should be able to move between two grid-cells within two tracking steps. Ultimately, a moderate increase in the tracking frequency to 1 h already results in significant improvements in tracking accuracy. A more in-depth investigation of the benefits of higher tracking frequencies of the tracking algorithm is provided in section 5.

2.3 Dynamic output

To take advantage of Prometheus in terms of file sizes, the ICON output facilities need to be adapted to allow for output domains to be dynamically restricted to small area(s) surrounding the tracked TC(s). The generic sub-module created for this purpose can be used in two ways:

- supplied file with prescribed output windows and times (off-line)
- connect to Prometheus (on-line)

The first option is meant for general use and debugging of the module. A sample file for how to use the module in this mode can be obtained from the linked gitlab repository. The second option generates output in a given radius around the TC centre(s)

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

tracked at the current time-step. The centres are collected in the tracking module as a global (entire tracking domain) array of coordinates, which is collected by the dynamic output module.

The module works by selecting all grid-centres whose coordinates are within the given radius around the TC centre. This is done for all provided TC centres resulting in a set of output files for each detected TC. The output grid is then restricted to the grid indices that are selected. At the moment, dynamic output only supports variables that are located in the cells. Support for variables on edges and vertices can be added in the same fashion as for cells if required.

Performance wise the generation of a single dynamic region is faster than the output of the whole static region. However, if coupled with the TC tracking algorithm, the number of dynamic regions output per output time step is around 25 – the exact number depends on TC activity and tracking criteria. This is because all systems that can potentially develop into a TC have to be taken into account. The time taken for output scales linearly with the number of dynamic output regions. To avoid an explosion in runtime, the output around TCs that have not yet fulfilled the minimum lifetime criterion are merged into one combined grid. This adds an additional small post processing step, but ensures the designated output PE does not stall the simulation by taking longer for output generation than the worker PEs for the time integration to the next output time step.

3 Numerical experiments setup

The underlying numerical model used for testing is ICON (Zängl et al., 2015, ICON partnership (DWD; MPI-M; DKRZ; KIT; C2SM) (2024). ICON release 2024.10. World Data Center for Climate (WDCC) at DKRZ). Simulations are run in limited area mode (LAM) with the simulation domain stretching from 120°W/0°N to 15°W/40°N. The grid used is an icosahedral grid with refinement of R02B09, resulting in cells with a diameter of ca. $5\,\mathrm{km}$. Vertically the domain is divided into 80 levels using SLEVE coordinates (Schär et al., 2002). The simulation time-step is set to 25 seconds and uses ICONs NWP physics package for parametrizations and the 2-moment cloud microphysics scheme (Seifert and Beheng, 2006). The model is initialized with IFS data which also provides sea-surface temperature and boundary data. The simulations start on August 1st, 2005 and end on October 31st, 2005, which is the most active part of the TC season, with four of the five category 5 hurricanes of 2005.

The tracking algorithm is used with the threshold parameters set to the values shown in table 1 with an additional mean sea level pressure threshold of $995\,\mathrm{hPa}$ applied to filter out very low intensity systems that are usually false positives. For the validation of Prometheus, its tracks are compared with those generated by the Epimetheus algorithm. For this the tracking time-step is set to $6\,\mathrm{h}$, with other parameter settings chosen to be the same for both algorithms. The tracks are mainly compared on a qualitative basis, as the sample size is too small for proper quantitative analysis, with the accumulated cyclonic energy (ACE) used as a quantitative sanity check to ensure that the tracking differences are confined to weak systems.

For the discussion of the higher tracking frequency (HTF), Prometheus is run with a tracking time-step set to $1\,\mathrm{h}$, hereafter referred to as HTF-P. Initial HTF-P simulations were done with the same parameter settings as Epimetheus, but later were adapted to get better results. A closer look at those adaptations is presented in section 5. The comparison between the HTF-P and the Epimetheus tracks is done quantitatively by assessing the average added accurate tracking steps, and qualitatively by showing examples of phenomena that can be captured with more precision.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

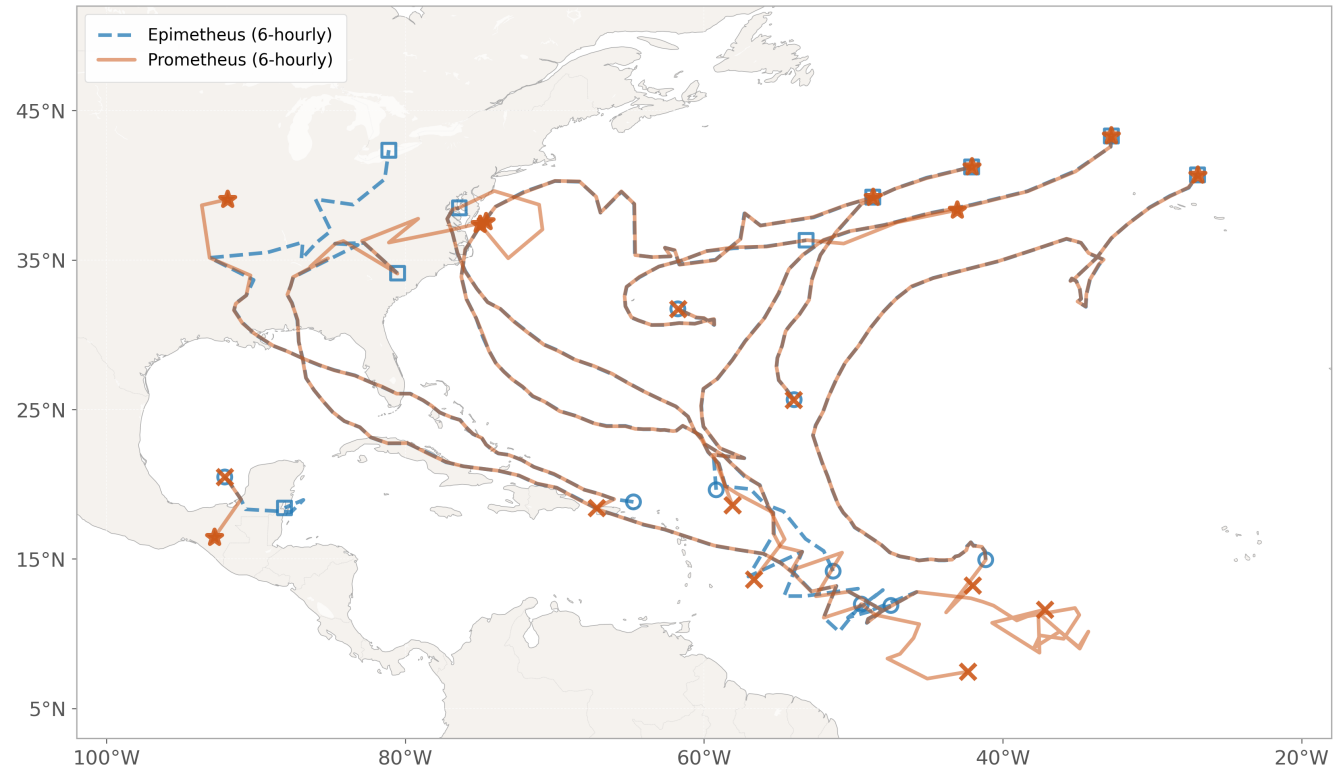


Figure 4. TC tracks using the Prometheus (Orange) Epimetheus (Blue, Dashed) tracking algorithms. The $\times, o$ denote the respective starting points and the $\star, \square$ denote the respective track terminations.

4 Comparing Epimetheus and Prometheus

The tracks produced between Epimetheus and Prometheus – using the same threshold values and tracking frequency – largely agree, as is expected due to the fact that their detection and stitching mechanisms are very similar. In Figure 4 we can observe some disagreements in the early phases and termination of some tracks. The figure only shows tracks that are detected by both algorithms. There are some weaker systems that are only detected by one of the algorithms, which generally are false positives caused by independent pressure minima being connected. There are two main reasons for the differences in tracks: the interpolation onto a regular grid for the off-line algorithm and differences in the effective search radii.

The interpolation can lead to small differences between the mean sea level pressure fields. During TC genesis, where the pressure minimum is often not as pronounced, this can lead to different local pressure minima to be chosen by the algorithms. During the initial phase, multiple vorticity maxima belonging to separate pressure minima coalesce into one cyclonic structure. Until then the tracking algorithms are prone to jumping between them and even small interpolation errors can change which minimum is tracked. An example of this can be found in the appendix (Appendix A, Track 1).

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

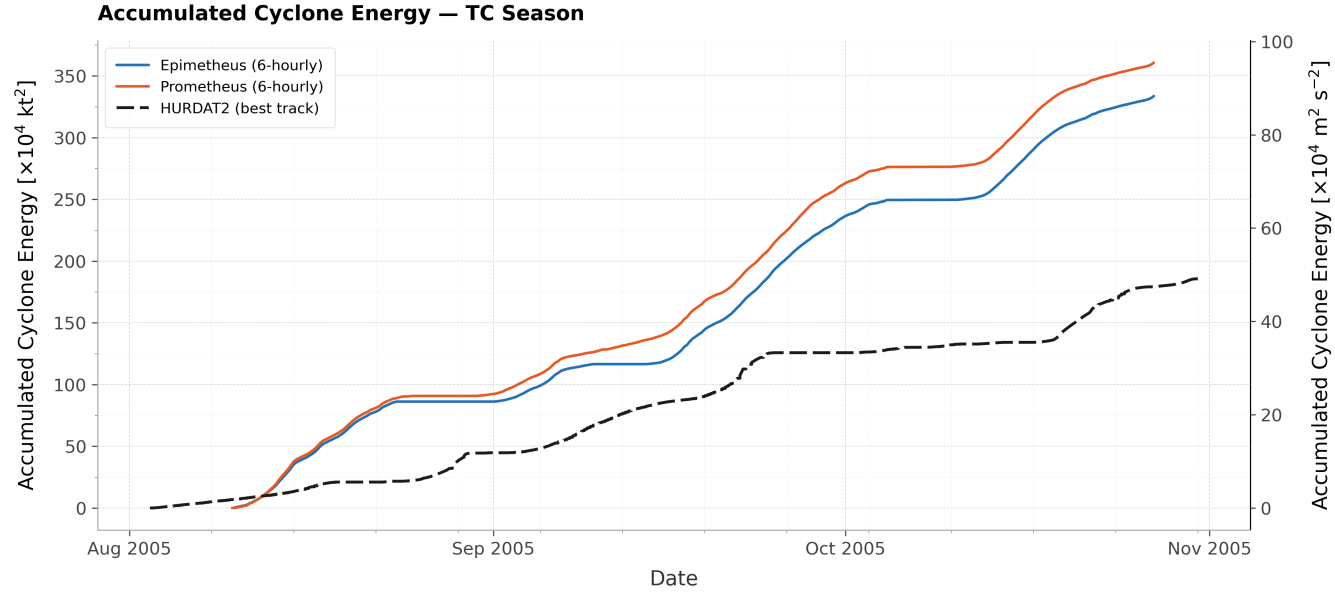


Figure 5. Accumulated cyclonic energy over the TC season, from different tracking algorithms.

The second difference is in the search radii used for the pressure criterion. The radius used in the off-line algorithm is based on a radians-per-meter calculation done at a reference latitude. This translates the search radius into a search angle. The resulting shape of the physical search area goes from circular at the equator to elliptical at higher latitudes. The total area also shrinks with higher latitudes, as the same longitude angle covers a smaller distance at higher latitudes. In terms of strictness of the pressure criterion this results in a more strict criterion at the equator, where the pressure has to be minimal within a larger area compared to at higher latitudes where the criterion becomes looser.

Despite those differences the tracks at higher intensities (category 1 and higher) are in almost complete agreement, which also can be verified by looking at the ACE in Figure 5, which match closely. In this case Prometheus captures a high latitude false positive with strong wind speeds, resulting in a somewhat higher ACE.

$$\mathrm{ACE} = 10^{-4} \sum v_{max}^2 \text{ (for } v_{max} \geq 17\,\mathrm{m\,s^{-1}} \approx 34\,\mathrm{kn}\text{), at } 6\,\mathrm{h} \text{ intervals} \tag{1}$$

5 Higher tracking frequencies

Increasing the tracking frequency to 1-hourly yields a large increase in tracking precision. It also effectively eliminates false positive tracks generated due to stitching together of two unrelated pressure minima. This happens because the only heuristic criterion indicating suitable stitching candidate is given by the assumed maximum translational velocity (MTV) of a TC of $20\,\mathrm{m\,s^{-1}}$. At a $6\,\mathrm{h}$ tracking frequency the allowed radius of movement is $432\,\mathrm{km}$. Within this radius there is a good chance for

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

two unrelated pressure minima existing that both fulfil also the other criteria coincidentally. This radius is reduced to 72 km for hourly tracking frequency.

With the added precision of the higher tracking frequency however, increased strictness is also introduced due to the smaller movement radius and higher rate of criteria checking. Using the tracker in the same setup as was used for the 6-hourly tracking frequency, results in cut-up tracks that are generally starting late and terminating early. Early termination is mainly the cause of increased noise in the translational velocity, due to centre oscillations and the discrete nature of the grid. Those tend to average out over the larger 6-hourly tracking frequency, but have to be considered at hourly frequencies. A simple way of taking the increased noise into account, is to increase the allowed MTV to $35\,\mathrm{m\,s^{-1}}$ at hourly tracking frequency.

The late genesis and the broken up tracks are caused by the higher rate of checking, as the criteria need to be fulfilled at every tracking step. While the pressure minimum and the positive vorticity remain relatively consistent over time, the temperature anomaly at 300 hPa is often less stable. This causes the warm core criterion to temporarily not be fulfilled.

5.1 Relaxation of warm core criterion

A close look at broken up tracks of Prometheus in hourly tracking frequency revealed that the warm core criterion is temporarily not fulfilled, causing the track detection to stop, although the TC is still there. This can be the case, because the 300 hPa temperature maximum is sometimes horizontally slightly off-set as compared to the pressure minimum. In Wang and Jiang (2019) it is stated that the warm core of weaker systems tends to reside somewhat lower at around 400 hPa. With the algorithm only checking at 300 hPa, this can lead to the warm core not being captured sufficiently. One example is illustrated in Figure 6. There a TC makes landfall over Cuba where it weakens substantially, but then starts to re-intensify. Initially the warm core is still detectable at 300 hPa albeit horizontally somewhat off-set. During the next hours the TC generates a new warm core at lower latitudes, which cannot be detected at 300 hPa. Once the TC intensifies enough, the warm core moves upward and becomes detectable at 300 hPa again. There are two approaches to relax the warm core criterion in order to track the system continuously: continue to track the system even if it temporarily "loses" its warm core, or allow for a range of pressure levels that are used to check the warm core criterion. In our implementation, both options can also be combined.

For the first approach the warm core criterion can remain unfulfilled up to a specified time under the restriction that all the other parameters need to pass all their thresholds. Nevertheless, the tracked system has to have had a warm core detected prior to this exception being allowed. The amount of time the warm core is allowed to be "lost" can be set as a parameter in the namelist. To most closely resemble the 6 h on- or off-line simulations, a setting of 6 h lost core duration is ideal, as this effectively reverses some of the added strictness from the higher tracking frequency. At a setting of 12 h all the broken up tracks from the 6-hourly tracking frequency were able to be connected in our test cases.

The pressure levels that should be used to check the warm core criterion in the second approach, can also be specified in the tracker's namelist. If a TC fulfils the criterion on at least one of them, it fulfils the overall criterion. This has the benefit of having a physical reasoning underlining it, and only stops the tracking as soon as the warm core disappears at these multiple pressure levels. It also is more generous for the first detection, resulting in additional detected tracks of lower intensity systems.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

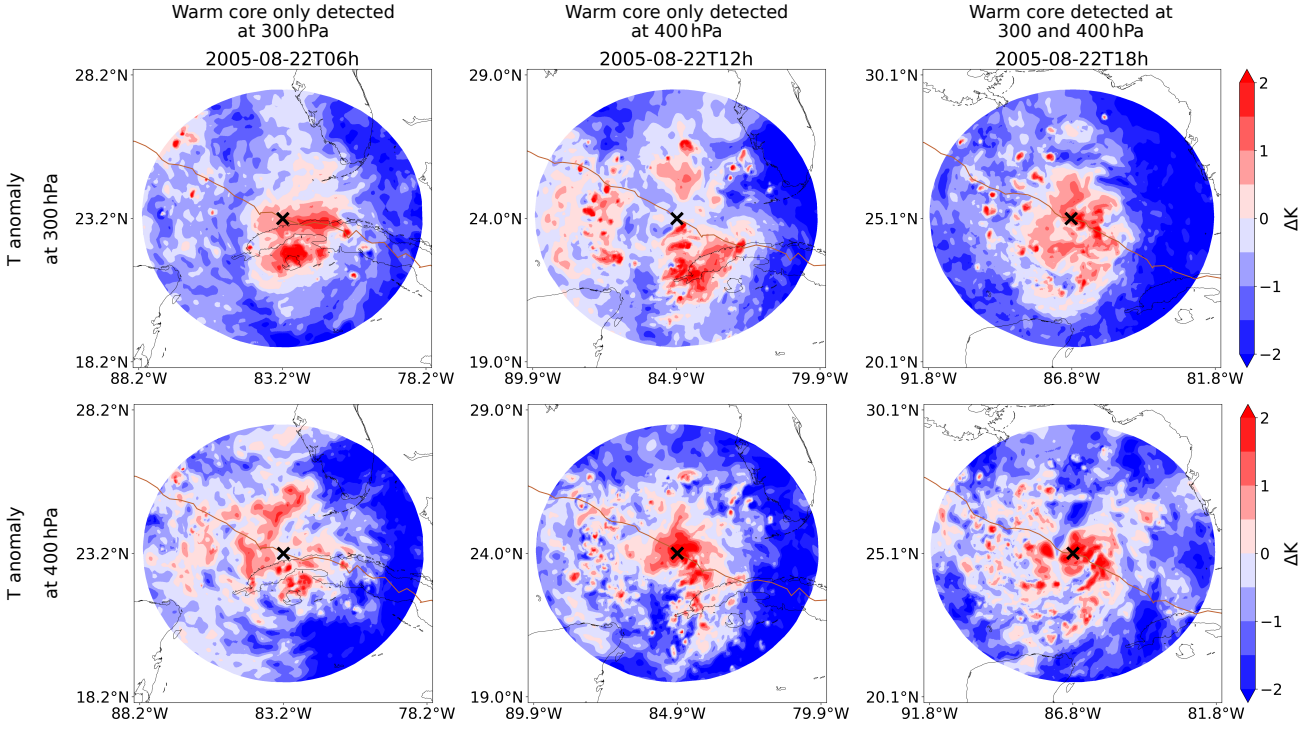


Figure 6. Plot of the warm core of a category 1 Hurricane (orange x) at three different time steps. The upper row shows the temperature anomaly at 300 hPa, the lower row at 400 hPa. By only looking at the one row the warm core can not be detected at every timestep.

Looking at the resulting tracks, it becomes clear that the loosening of the warm core criterion leads to an increase in both true detections but also false positives mostly in the form of extratropical cyclones. In order to benefit from the longer tracks possible with relaxed criteria, while keeping or reducing the number of false positives, different strictness levels can be chosen during the post processing of the tracks:

- no restriction: any TC point tracked using the relaxed warm core criteria counts toward the minimum lifetime.
- semi-strict: TC points tracked using the relaxed warm core criteria do not count toward the minimum lifetime.
- strict: TC needs to be detected using the non-relaxed warm core criteria in consecutive tracking steps until the minimum lifetime is reached. (This yields no additional tracks compared to the standard simulation, but longer ones)

Using the semi-strict criteria represents a good compromise between avoiding detection of extratropical cyclones and better capturing of weaker systems and is the setting used for the comparisons in the rest of this paper.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

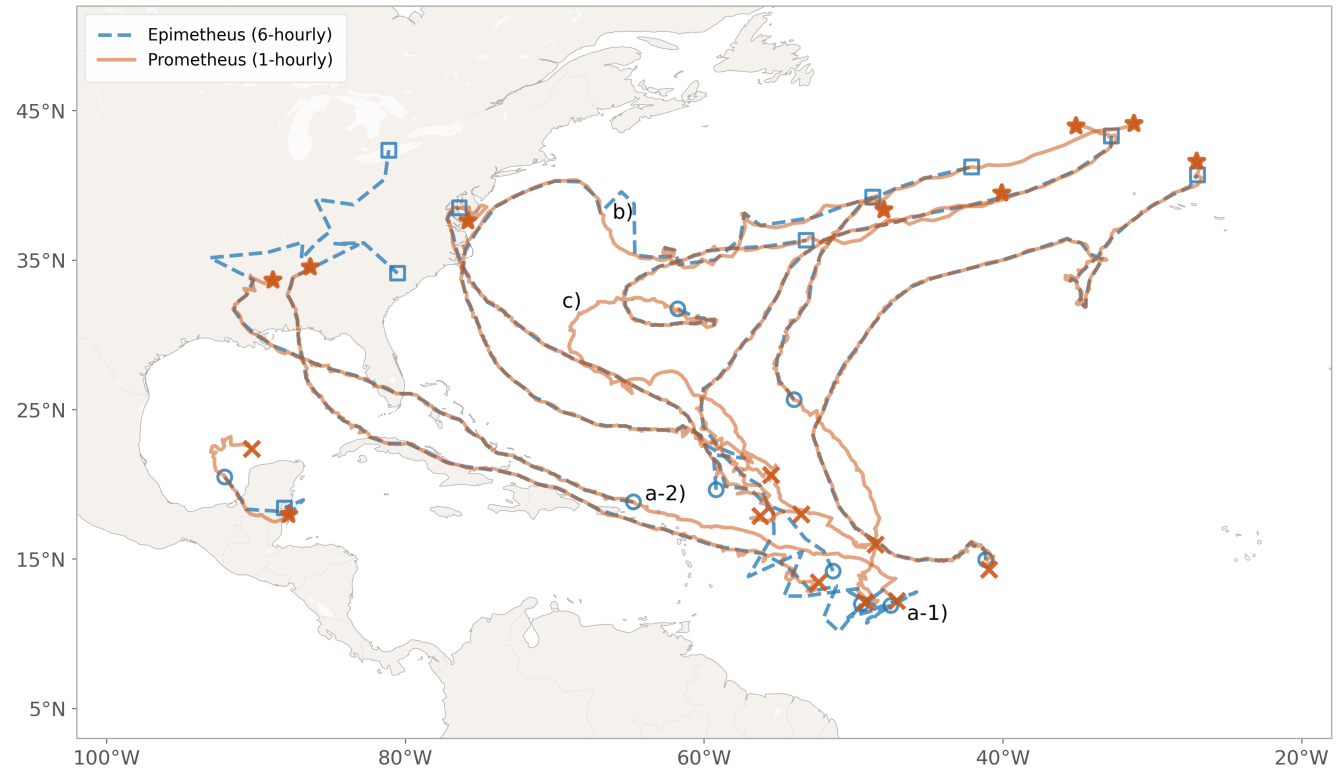


Figure 7. TC tracks of Prometheus using the HTF LC/WC$_{lower}$ setup (orange) and Epimetheus (Blue, Dashed) tracking algorithms. The $\times, o$ denote the respective starting points and the $\star, \square$ denote the respective track terminations.

5.2 Track improvements

The more flexible warm core criterion together with the smaller movement radius due to the higher tracking frequency leads to both more exact, but also more extensive tracks, while at the same time reducing false detections.

The main improvements of the HTF version are seen in the tropical depression and tropical storm phases of the tracks. This can be seen in Figure 7, where the HTF Prometheus (HTF-P) tracks – using a combination of $12\,\mathrm{h}$ lost core (LC) and warm core (WC) pressure levels between $300\,\mathrm{hPa}$ and $400\,\mathrm{hPa}$ (LC/WC$_{lower}$) – are plotted against the 6-hourly Epimetheus (EPI) tracks. The improvements are specifically the TC genesis, where for some tracks the higher tracking frequency reveals a much earlier initial position of the system, such as track (a). This is because at the beginning of the tracks the pressure minima are not yet as pronounced and the tracking is more prone to jump between unrelated pressure minima. Reducing the permissible movement radius reduces the risk of this tracking error and the resulting spurious beginnings of the tracks.

Also phases of low intensity in the middle of the track are tracked more precisely thanks to the flexible warm core criterion. An example for this is shown in track (b), where the Epimetheus tracker stitches to an unrelated and weaker pressure minimum

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

Figure 8. TC undergoing extra-tropical transition. The transition is captured by using the "lost core" feature.

that fulfils the warm core criterion by chance. The addition of the lost-core option also means that tracks that contained gaps in the 6 h Prometheus run can now be connected. One example is track (c).

The higher tracking frequency also allows for more precise tracking of commonly studied TC phenomena. One of which is the eye-wall replacement cycle of a rapidly intensifying TC, which may only take 12 h (Fischer et al., 2020). In such a case it is beneficial to have more than 2 or 3 data points to study. Another improvement is in the landfall detection. Landfall usually causes a TC to gradually decay. The HTF version can detect time and location of landfall more precisely.

A welcomed side-effect of the more flexible warm core criterion is that the termination of TCs can now be plotted with more detail as the system will be tracked for some time-steps after the warm core has vanished, which is the most common reason for termination in our simulation. The loss of the warm core, which is usually caused by higher wind shear and colder sea-surface temperature over the ocean at higher latitudes, weakens the TC (Jones et al., 2003). Some cyclones then rapidly intensify again and transition into an extra-tropical cyclone signified by a frontal structure and a cold core (Jones et al., 2003). Thanks to the option to allow for the absence of the warm core for some time, this process can be better analysed. Figure 8 showcases such an event. In the temperature panels, the weakening of the warm core can be observed, and the wind speed panels show the loss of symmetry. Both panels also show a transition to a frontal structure, typical for extra-tropical cyclones. The trajectory files produced by the on-line tracker indicate the added tracking points after the last warm core detection as extratropical cyclone to ensure definitionally correct TC tracks.

Overall, the HTF-P tracker with the flexible warm core criteria yields an average tracking improvement of +32.8 h in early genesis detection. Decay performance is largely similar, with HTF-P tracks terminating earlier after landfall. In these cases the tails of the 6-hourly EPI tracks are often erroneous, with at most one additional correct tracking step compared to HTF-P. Over the full track, seven of nine tracks captured by both trackers show improvements using the HTF-P tracker.

There are two instances, where the Epimetheus (and 6-hourly Prometheus) tracker detects the system earlier. In those cases the initial EPI tracks are spurious, jumping between the actual system and unrelated pressure minima.

6 File Sizes and Runtime

One of the main thrusts behind taking the algorithm on-line is the promise of reduced file sizes when combining the tracking algorithm with the dynamic output submodule. When choosing an output radius of 500 km around the tracked TC centres, the resulting file size is reduced by a factor of about 67 in the current setup. Important to note here is that the number of output files now depends on the number of tracked systems instead of the simulation time. Because for the majority of time, there are no TCs present in the basin, this further reduces storage burden.

To showcase the results a set of 20 atmospheric variables over 80 height levels is chosen. The resulting storage requirements of the full output is $10\,\mathrm{GB} \times 2160 \approx 21.6\,\mathrm{TB}$ for hourly output from August through October. This is prohibitive for doing

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

analysis on a local laptop. If we assume for the dynamic output a generous estimate of a TC present for half of the simulation time, the resulting storage requirement will be about: $150\,\mathrm{MB} \times 1080 \approx 162\,\mathrm{GB}$, which is much more manageable.

Lastly the runtime difference is worth considering. During development of Prometheus some efficiency improvements were made that can be found in appendix B. These could also be implemented for Epimetheus. Working on the unstructured and decomposed grid does incur additional cost due to the search algorithm and the MPI communication.

Running the simulation on 10 compute nodes on the Alps supercomputer (2880 CPU cores), the average runtime for one call to the tracking algorithm is about $460\,\mathrm{ms}$, which is about 1.3 times the average integration time for one time step. Increasing the node count has a negative impact on performance of the tracker, due to the increasing MPI overhead. At the coarse frequency with which the algorithm is called ($6\,\mathrm{h}$/every 1440th time step or $1\,\mathrm{h}$/every 360th time step), the runtime does not noticeably affect the overall runtime of the simulation.

Increasing the tracking frequency leads to a small performance improvement per call. This is mainly due to the reduced movement radius, which means that the potential to traverse to another domain part is reduced, resulting in fewer MPI calls per tracking iteration. Because those calls make up the majority of the runtime, a reduction in MPI calls has a significant effect on the total load. On average this amounts to a 10-15% decrease in runtime per call to the tracker.

In addition to the simulation runtime, there are also some necessary post-processing steps to take into account as some of the steps can only be done by considering past time steps. This includes deleting output for TCs that did not fulfil the lifetime- or minimum detection criteria. Time taken for this varies depending on output frequency, the number of output variables, strictness of criteria, etc. Ballpark for tracking one TC season is roughly 12 minutes for extracting the trajectories and optionally an additional 12 minutes to extract the dynamic output from the merged output (assuming hourly output for 10 TCs).

7 Tracking Errors

In sections 4 and 5 false positives in the tracked systems were mentioned. This section focuses on why they occur and what types of errors are particularly prone to occur depending on different parameter settings of the tracking algorithm. The error types can be roughly categorized into three different – but not mutually exclusive – causes. Often multiple error types can be observed within one false positive case, as is shown in multiple examples in the appendix (Appendix A, e.g. track 13).

1. Independent pressure minima are connected
2. The "system" is at the edge of a temperature front (temperature criterion fulfilled, other criteria fulfilled by chance, pressure criterion sometimes because of enclosing topography)
3. Tracking an extra tropical cyclone (ETC) (frontal structure)

The first error type can be avoided by using higher tracking frequencies (e.g. $1\,\mathrm{h}$). Although if two minima are very close this can still lead to jumps in tracks (see Appendix A, Track 13 genesis). Error type two is largely eliminated by stricter pressure

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

criterion radii since they are often only barely fulfilling the detection criteria. The first error type also often occurs at the first couple of detections of an ultimately correct track, when the pressure minimum is not yet pronounced enough. Zigzaging shapes at the beginning of tracks are a common symptom of such an error. These errors are the price for detecting tracks as early as possible with the tracking criteria being looser at genesis.

Accidentally tracking ETCs (see Appendix A, Track 8) is an often occurring issue with TC tracking algorithms and is also mentioned to be the main source of tracking errors in the off-line tracking algorithm (Enz et al., 2023). These can be reduced by increasing the minimum sea-surface temperature criterion at genesis, but this has the negative effect of not detecting tropical transition (TT) genesis, which takes place at higher latitudes and thus lower temperatures. Given that around 40% of Atlantic TCs originate from TT (Hart and Evans, 2001), ignoring them is not advisable. The last error can also occur at the end of a track, when TCs transition into a ETC and tracking sometimes continues to detect the transitioned system. This can of course also be seen as a welcome part of the tracker.

8 Conclusions

Tracking systems – such as TCs – inside the numerical model that simulates them, provides multiple advantages over running a separate off-line tracker. Prometheus enables more precise tracking by increasing the tracking frequency at little additional computational or data cost. Additionally through the dynamic output module, the created output file sizes can be reduced significantly, which will become more important with higher resolution models in the future where output file sizes will become even larger.

Using the same tracking frequencies, the tracks of Prometheus and Epimetheus overlap for the majority of their length. Differences in tracks are almost exclusively below hurricane intensity. The cause for the differences can be tracked down to the interpolation onto the regular grid for Epimetheus and the shape of the search area used for the criteria checking.

When using the higher tracking frequencies, the tracker becomes stricter because of higher checking frequency. This leads to more accurate results but also has the side effect of breaking up tracks. To counteract this the warm core criterion has been made more flexible, retaining the accuracy gains while keeping the tracks connected.

While Prometheus as presented here performs very well, there are three areas where future improvements could be made. The first is the dynamic output. Currently dynamic output is done without dedicated output PEs, leading to increased run-times, as the simulation is stalled during output generation. Multiple steps have already been taken to mitigate the overhead caused, but at present the overhead becomes limiting when running simulations with more than 20 Alps nodes (5760 CPU cores). This issue might disappear by itself when moving over to GPU simulations, where output generation is not a limiting factor.

The next idea concerns the frequency at which the tracking algorithm is called. While hourly tracking frequency yields accurate tracks, there might be cases where we want the highest possible temporal tracking resolution. To avoid causing a large tracking overhead, the criteria for pressure minimum, vorticity and temperature could only be checked at hourly intervals. At each simulation step however, the algorithm would check if the pressure minimum has moved to one of the surrounding cells, by evaluating the mean sea level pressure on the current cell as well as the surrounding "ring" of cells. The maximum

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

translational velocity of the TC as well as the Courant-Friedrichs-Lewy (CFL) criterion ensure that the system does not move beyond the surrounding "ring" of cells. One benefit for performance would be that cross border communication could be massively simplified – and MPI cost reduced – because the information of what PE border could potentially be crossed is given by the corresponding halo cells. Implementation would require a substantial redesign of the algorithm itself to accommodate for the two different tracking frequencies.

Another improvement can be pursued regarding extratropical cyclones. Currently the transition is captured by tracking the system after it loses the warm core, given that it remains strong enough to fulfil the other criteria. While the loss of warm core can be gathered from the track output, a more tailored approach could provide more insight into the ETT process – including onset and completion of ETT. One such approach was developed by Evans and Hart (2003) and proposes to use an asymmetry criterion for the onset of ETT and the formation of the cold core for the completion. The cold core is evaluated using the thermal wind.

Lastly the sea surface temperature criterion of the tracking algorithm could be adjusted to better detect TCs originating from tropical transition. McTaggart-Cowan et al. (2015) mentioned an extended criterion to allow for lower genesis SST if an additional "coupling threshold" is not exceeded. The proposed threshold is a maximum difference in equivalent potential temperature between tropopause and $850\,\mathrm{hPa}$ level of $22.5\,^{\circ}\mathrm{C}$. This additional threshold could be incorporated into the existing algorithm – also in the off-line version.

Our implementation of Prometheus as the on-line counterpart to the Epimetheus TC tracker allows for higher tracking frequency – resulting in a significant improvement in track precision – while at the same time reducing the output storage burden due to the dynamic output. This approach will likely gain in value when moving towards higher resolutions and sets an example how to efficiently reduce data amount while gaining tracking quality.

Code and data availability. The model code and data used in this paper is publicly available on Zenodo under https://doi.org/10.5281/zenodo.22301412 (Meier et al., 2026). This includes the Prometheus and Epimetheus source codes, the post processing and plotting scripts, as well as the data required to reproduce the plots in this paper. The open-source ICON model code version 2024.10 used for our simulations can be obtained at https://icon-model.org/. Additionally, the post processing and plotting scripts can also be accessed on the following GitLab repo: https://gitlab.ethz.ch/thimeier/python-scripts-for-tc-tracking/-/tree/main

Appendix A: Analysis of tracks in TC season 2005 (August-October)

This part of the appendix gives a quick qualitative assessment of each tracked system, describing if it is a true positive or what tracking errors can be observed. Additionally the genesis type and reason for termination are mentioned and a comparison made with its off-line counterpart (if exists). Since the higher tracking frequency has improved detection of weaker systems, increasing the maximum central pressure threshold, yields additional correct tracks. Plots of all tracked systems are provided in the link at the end of the paper.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

Tracks from 1 h tracking frequency runs (HTF, see Figure A1), using HTF LC/WC$_{lower}$ setup. Only pressure minima below 1000 hPa are taken:

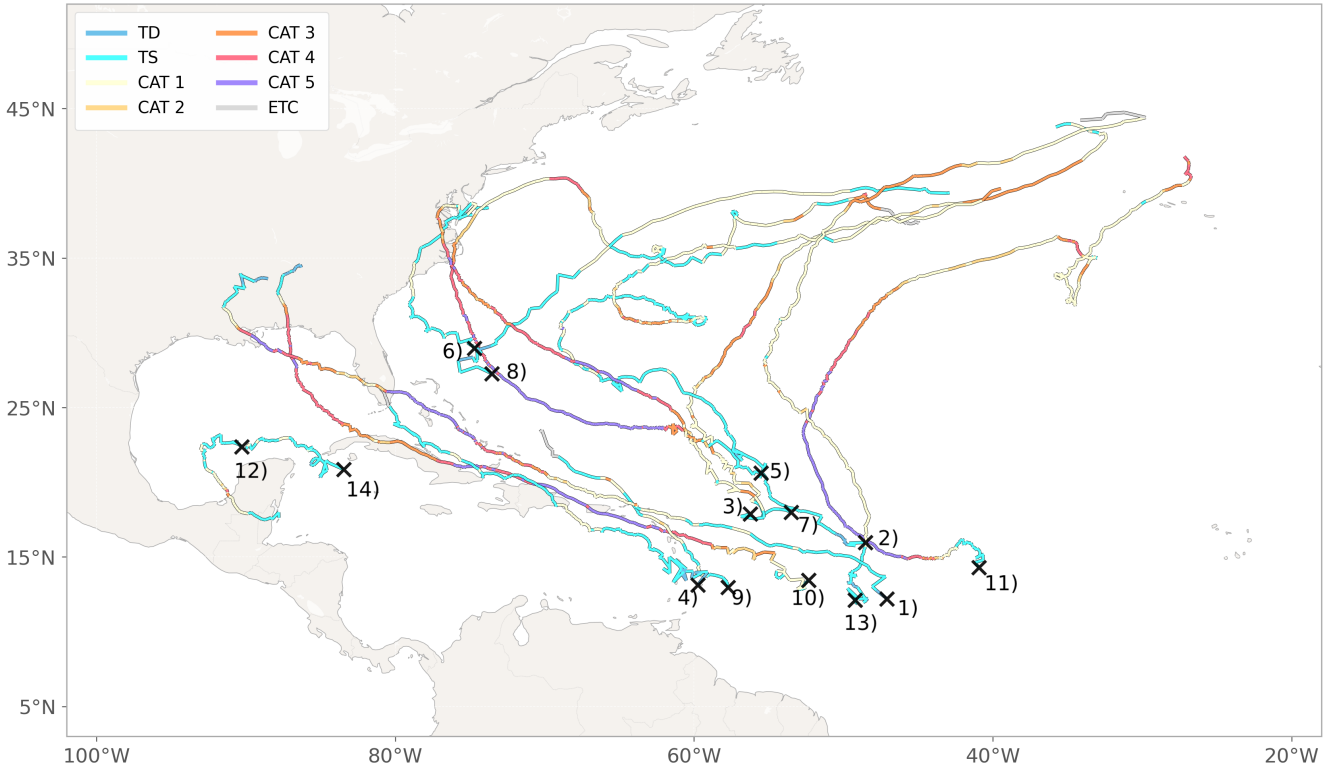


Figure A1. All TC Tracks of simulated TC season, using HTF LC/WC$_{lower}$ setup of the Prometheus tracking algorithm.

- Track 1: HTF-P has 67 h earlier detection, landfall track more accurate
- Track 2: HTF-P has 119 h earlier detection, termination similar
- Track 3: EPI has 36 h earlier detection (spurious, jumps between different pressure minima), HTF-P has 15 h later termination
- Track 4: HTF-P only true detection
- Track 5: HTF-P has 42 h earlier detection and 14 h later termination
- Track 6: HTF-P only true detection (>995 hPa)
- Track 7: EPI has 15 h earlier detection. EPI features two erroneous tracking steps (12 h) in the middle
- Track 8: HTF-P only, edge-case ETC (short lived warm core, >995 hPa)

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

- Track 9: HTF-P only true detection (>995 hPa)
- Track 10: genesis similar, termination after landfall more accurate in HTF-P
- Track 11: genesis and termination similar
- Track 12: HTF-P has 39 h earlier detection, more accurate track
- Track 13: HTF-P has 80 h earlier detection, similar termination
- Track 14: HTF-P only true detection (first part of track 12, >995 hPa)

Author contributions. The implementation of the on-line tracking algorithm and the provided scripts was done by TM. Implementation of dynamic output was done by TM with improvements made by IB. Code review was done by SF. The manuscript was written by TM and reviewed and edited by AC, BE, UL and SF.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The computational resources for the simulations and data processing were provided by Swiss National Supercomputing Centre (CSCS, project ID lp88). Microsoft Copilot and Gemini were used for partial proofreading of this manuscript and GitHub Copilot was used for editing the plotting and post processing scripts.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.

References

Bourdin, S., Fromang, S., Dulac, W., Cattiaux, J., and Chauvin, F.: Intercomparison of four algorithms for detecting tropical cyclones using ERA5, Geoscientific Model Development, 15, 6759–6786, https://doi.org/10.5194/gmd-15-6759-2022, 2022.

Bourdin, S., Hodges, K., Han, Y., Saffin, L., Baker, A., Methven, J., and Vidale, P. L.: RABTracks: Reanalysis-Augmented Best Tracks for Tropical Cyclones, Earth System Science Data Discussions, 2026, 1–32, https://doi.org/10.5194/essd-2026-403, 2026.

Camargo, S. J. and Zebiak, S. E.: Improving the Detection and Tracking of Tropical Cyclones in Atmospheric General Circulation Models, Weather and Forecasting, 17, 1152 – 1162, https://doi.org/10.1175/1520-0434(2002)017<1152:ITDATO>2.0.CO;2, 2002.

Enz, B. M., Engelmann, J. P., and Lohmann, U.: Use of threshold parameter variation for tropical cyclone tracking, Geoscientific Model Development, 16, 5093–5112, https://doi.org/10.5194/gmd-16-5093-2023, 2023.

Evans, J. L. and Hart, R. E.: Objective Indicators of the Life Cycle Evolution of Extratropical Transition for Atlantic Tropical Cyclones, Monthly Weather Review, 131, 909 – 925, https://doi.org/10.1175/1520-0493(2003)131<0909:OIOTLC>2.0.CO;2, 2003.

Fischer, M. S., Rogers, R. F., and Reasor, P. D.: The Rapid Intensification and Eyewall Replacement Cycles of Hurricane Irma (2017), Monthly Weather Review, 148, 981 – 1004, https://doi.org/10.1175/MWR-D-19-0185.1, 2020.

Galea, D., Kunkel, J., and Lawrence, B. N.: TCDetect: A New Method of Detecting the Presence of Tropical Cyclones Using Deep Learning, Artificial Intelligence for the Earth Systems, 2, e220 045, https://doi.org/10.1175/AIES-D-22-0045.1, 2023.

Galea, D., Hodges, K., and Lawrence, B. N.: Investigating Differences between Tropical Cyclone Detection Systems, Artificial Intelligence for the Earth Systems, 3, e220 046, https://doi.org/10.1175/AIES-D-22-0046.1, 2024.

Hart, R. E. and Evans, J. L.: A Climatology of the Extratropical Transition of Atlantic Tropical Cyclones, Journal of Climate, 14, 546 – 564, https://doi.org/10.1175/1520-0442(2001)014<0546:ACOTET>2.0.CO;2, 2001.

Hodges, K. I.: A General Method for Tracking Analysis and Its Application to Meteorological Data, Monthly Weather Review, 122, 2573 – 2586, https://doi.org/10.1175/1520-0493(1994)122<2573:AGMFTA>2.0.CO;2, 1994.

Hodges, K. I.: Adaptive Constraints for Feature Tracking, Monthly Weather Review, 127, 1362 – 1373, https://doi.org/10.1175/1520-0493(1999)127<1362:ACFFT>2.0.CO;2, 1999.

Jones, S. C., Harr, P. A., Abraham, J., Bosart, L. F., Bowyer, P. J., Evans, J. L., Hanley, D. E., Hanstrum, B. N., Hart, R. E., Lalaurette, F., Sinclair, M. R., Smith, R. K., and Thorncroft, C.: The Extratropical Transition of Tropical Cyclones: Forecast Challenges, Current Understanding, and Future Directions, Weather and Forecasting, 18, 1052 – 1092, https://doi.org/10.1175/1520-0434(2003)018<1052:TETOTC>2.0.CO;2, 2003.

McTaggart-Cowan, R., Davies, E. L., Fairman, J. G., Galarneau, T. J., and Schultz, D. M.: Revisiting the 26.5°C Sea Surface Temperature Threshold for Tropical Cyclone Development, Bulletin of the American Meteorological Society, 96, 1929 – 1943, https://doi.org/10.1175/BAMS-D-13-00254.1, 2015.

Meier, T., Ferrachat, S., Enz, B., and Bouzdine, I.: Prometheus – Code and Data, https://doi.org/10.5281/zenodo.22301412, 2026.

NOAA: Hurricane costs, https://coast.noaa.gov/states/fast-facts/hurricane-costs.html – last accessed: Feb. 1st. 2024, 2024.

Prill, F., Reinert, D., Rieger, D., and Zängl, G.: ICON Tutorial - Working with the ICON Model, https://www.dwd.de/DE/leistungen/nwv_icon_tutorial/pdf_einzelbaende/icon_tutorial2023.pdf;jsessionid=3BE2346701DC3F1FABB0D67EA0FD3B2B.live21064?__blob=publicationFile&v=3, 2023.

https://doi.org/10.5194/egusphere-2026-5568
Preprint. Discussion started: 12 November 2026
© Author(s) 2026. CC BY 4.0 License.



Schär, C., Leuenberger, D., Fuhrer, O., Lüthi, D., and Girard, C.: A New Terrain-Following Vertical Coordinate Formulation for Atmospheric Prediction Models, Monthly Weather Review, 130, 2459 – 2480, https://doi.org/10.1175/1520-0493(2002)130<2459:ANTFVC>2.0.CO;2, 2002.

Seifert, A. and Beheng, K. D.: A two-moment cloud microphysics parameterization for mixed-phase clouds. Part 1: Model description, Meteorology and Atmospheric Physics, 92, 45–66, https://doi.org/10.1007/s00703-005-0112-4, 2006.

Senf, F. and Cremer, R.: Advancing the capabilities for efficient hurricane-centric simulations with the atmospheric model ICON, Geoscientific Model Development, 19, 7705–7723, https://doi.org/10.5194/gmd-19-7705-2026, 2026.

Tory, K. J., Dare, R. A., Davidson, N. E., McBride, J. L., and Chand, S. S.: The importance of low-deformation vorticity in tropical cyclone formation, Atmospheric Chemistry and Physics, 13, 2115–2132, https://doi.org/10.5194/acp-13-2115-2013, 2013.

Ullrich, P. A. and Zarzycki, C. M.: TempestExtremes: a framework for scale-insensitive pointwise feature tracking on unstructured grids, Geoscientific Model Development, 10, 1069–1090, https://doi.org/10.5194/gmd-10-1069-2017, 2017.

Ullrich, P. A., Zarzycki, C. M., McClenny, E. E., Pinheiro, M. C., Stansfield, A. M., and Reed, K. A.: TempestExtremes v2.1: a community framework for feature detection, tracking, and analysis in large datasets, Geoscientific Model Development, 14, 5023–5048, https://doi.org/10.5194/gmd-14-5023-2021, 2021.

Wang, X. and Jiang, H.: A 13-Year Global Climatology of Tropical Cyclone Warm-Core Structures from AIRS Data, Monthly Weather Review, 147, 773 – 790, https://doi.org/10.1175/MWR-D-18-0276.1, 2019.

Zarzycki, C. M. and Ullrich, P. A.: Assessing sensitivities in algorithmic detection of tropical cyclones in climate data, Geophysical Research Letters, 44, 1141–1149, https://doi.org/https://doi.org/10.1002/2016GL071606, 2017.

Zängl, G., Reinert, D., Rípodas, P., and Baldauf, M.: The ICON (ICOsahedral Nonhydrostatic) modelling framework of DWD and MPI-M: Description of the nonhydrostatic dynamical core, Quarterly Journal of the Royal Meteorological Society, 141, https://doi.org/10.1002/qj.2378, 2015.