Channel Adjacency Algorithm (Horizontal Muon Algorithm refinement)

Simranjit Singh Chhibra (QMUL)

Dune DAQ DS/PP meeting, 3 April 2024
https://indico.fnal.gov/event/64132/

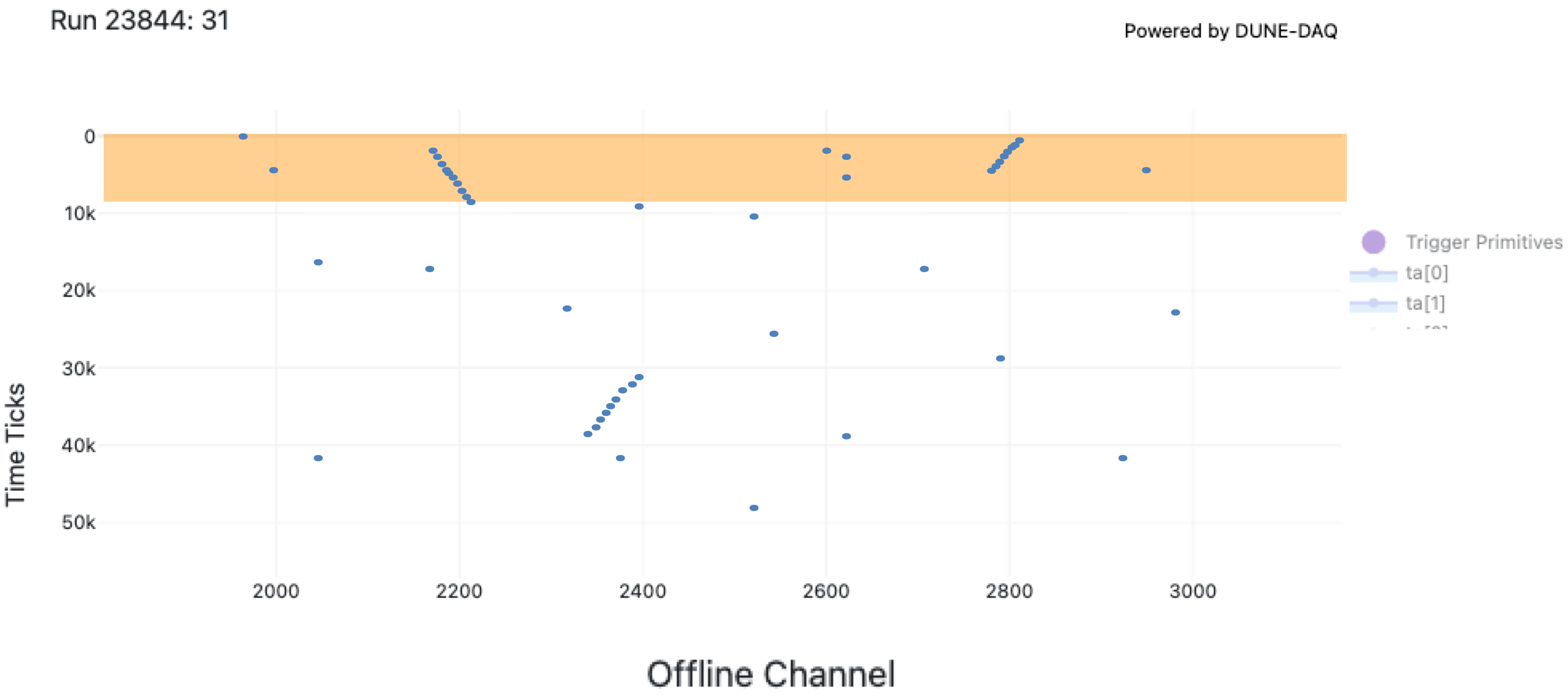
Technicalities

- **Input Data:**
  - **01/24 coldbox (VD):** np02vdcoldbox_tp_**run023844**_0000_tpwriter_tpswriter_*.hdf5

- **Horizontal Muon TA (HMTA) algorithm:**
  - trigger_on_adjacency = true
  - adjacency_threshold = {20, 50} *//min no. of adj channels to form a TA*
  - adj_tolerance = 5 *//total gap allowed (max gap allowed = 5)*
  - window_length = 8000 *//one clock tick = 16 ns*

HMTA algorithm

- TPs within 8k ticks make a TP window
- Check for activities (eg, adjacency_threshold = 8; adj_tolerance = 2)
- If activity found (True in this case) → build a TA

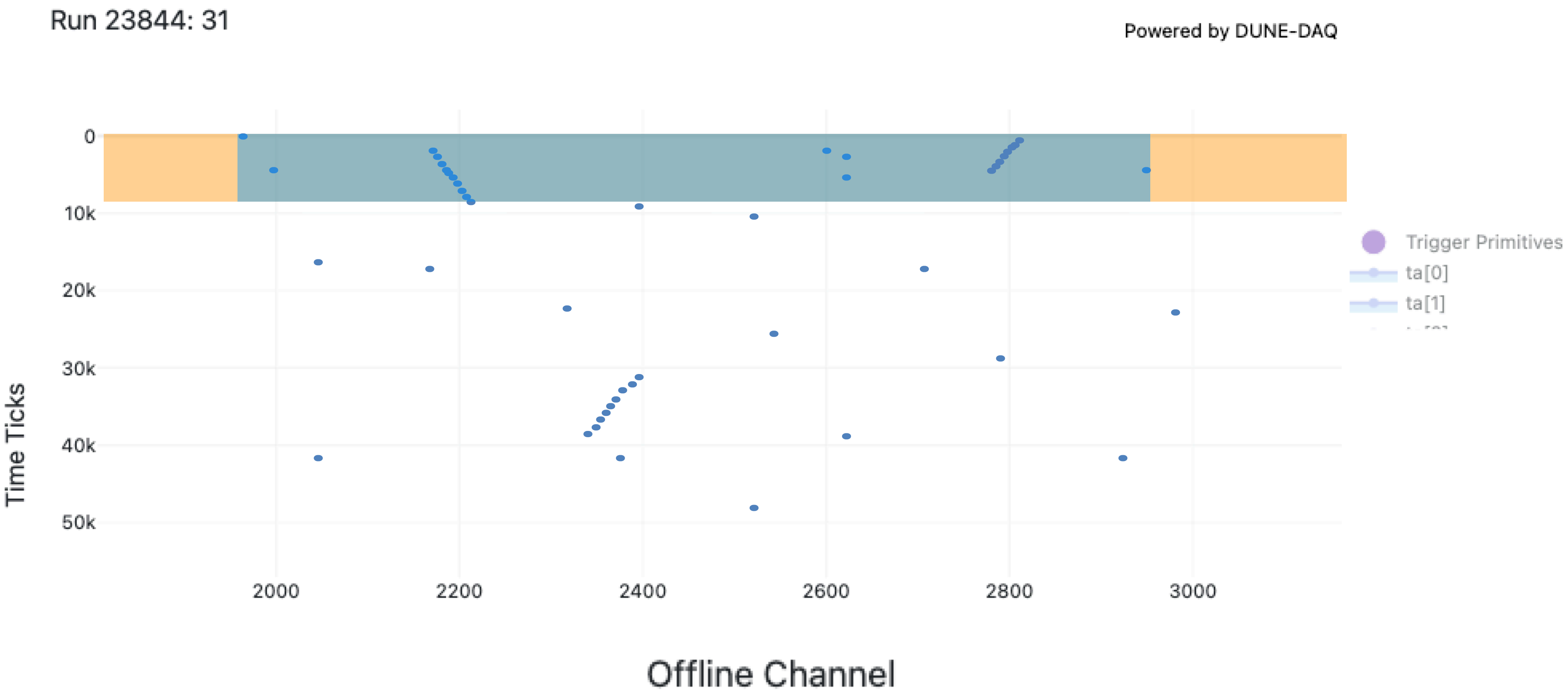


For illustration purpose!

HMTA algorithm

- **TA boundaries**
- **Y-axis:** time_start = time of window's first TP
  time_end = time of window's last TP
- **X-axis:** first_channel = lowest channel among all window TPs
  last_channel = highest channel among all window TPs

- TPs within 8k ticks make a TP window
- Check for activities (eg, adjacency_threshold = 8; adj_tolerance = 2)
- If activity found (True in this case) → build a TA

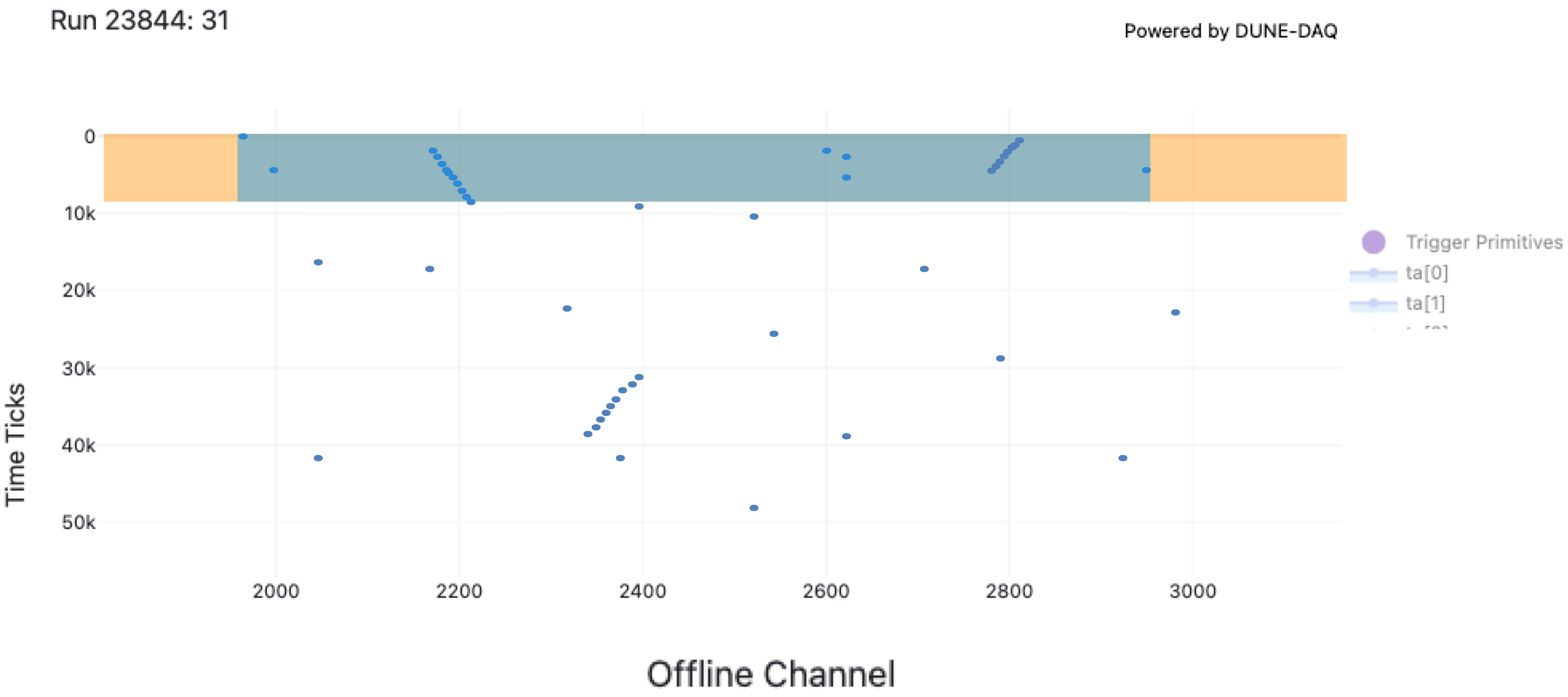


For illustration purpose!

HMTA algorithm

- **TA boundaries**
- **Y-axis:** time_start = time of window's first TP
  time_end = time of window's last TP
- **X-axis:** first_channel = lowest channel among all window TPs
  last_channel = highest channel among all window TPs

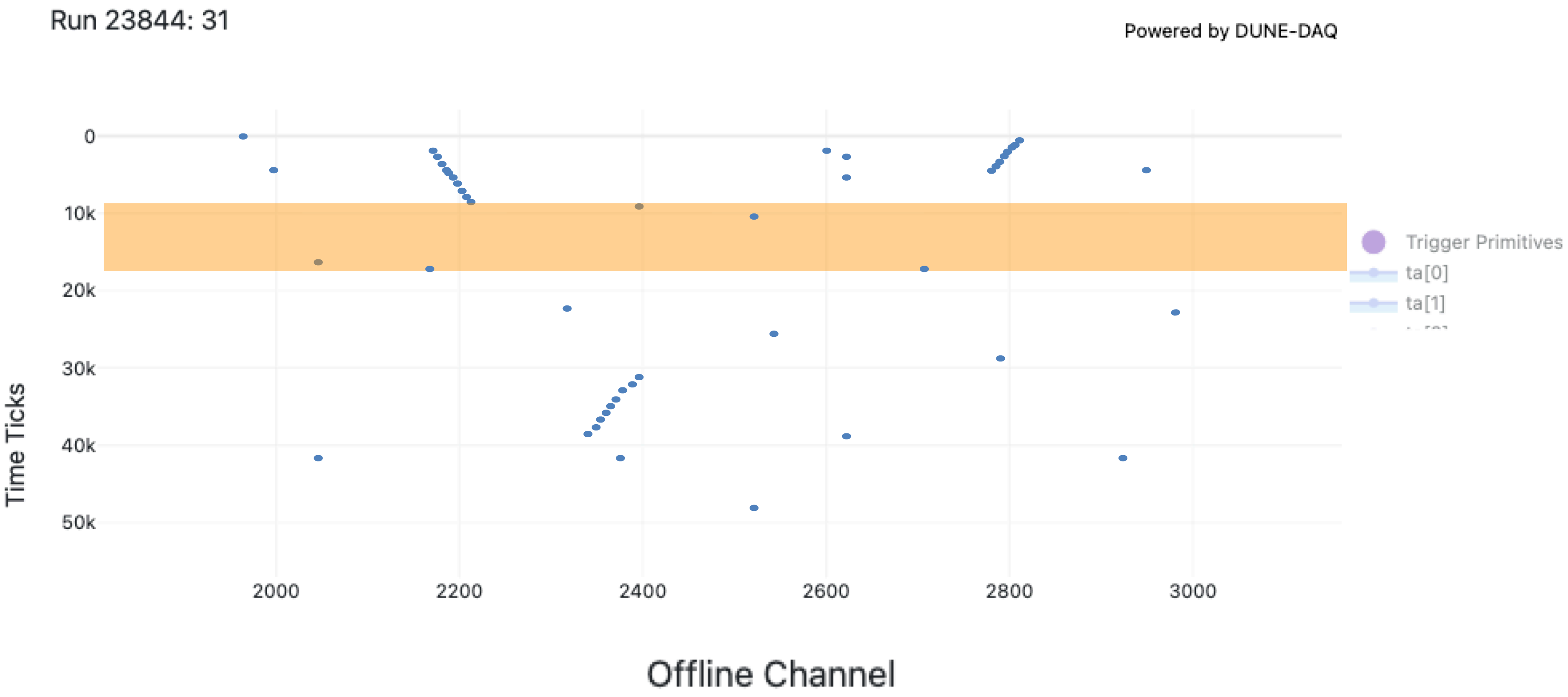
- TPs within 8k ticks make a TP window
- Check for activities (eg, adjacency_threshold = 8; adj_tolerance = 2)
- If activity found (True in this case) → build a TA → reset the TP window (next slide)



For illustration purpose!

HMTA algorithm

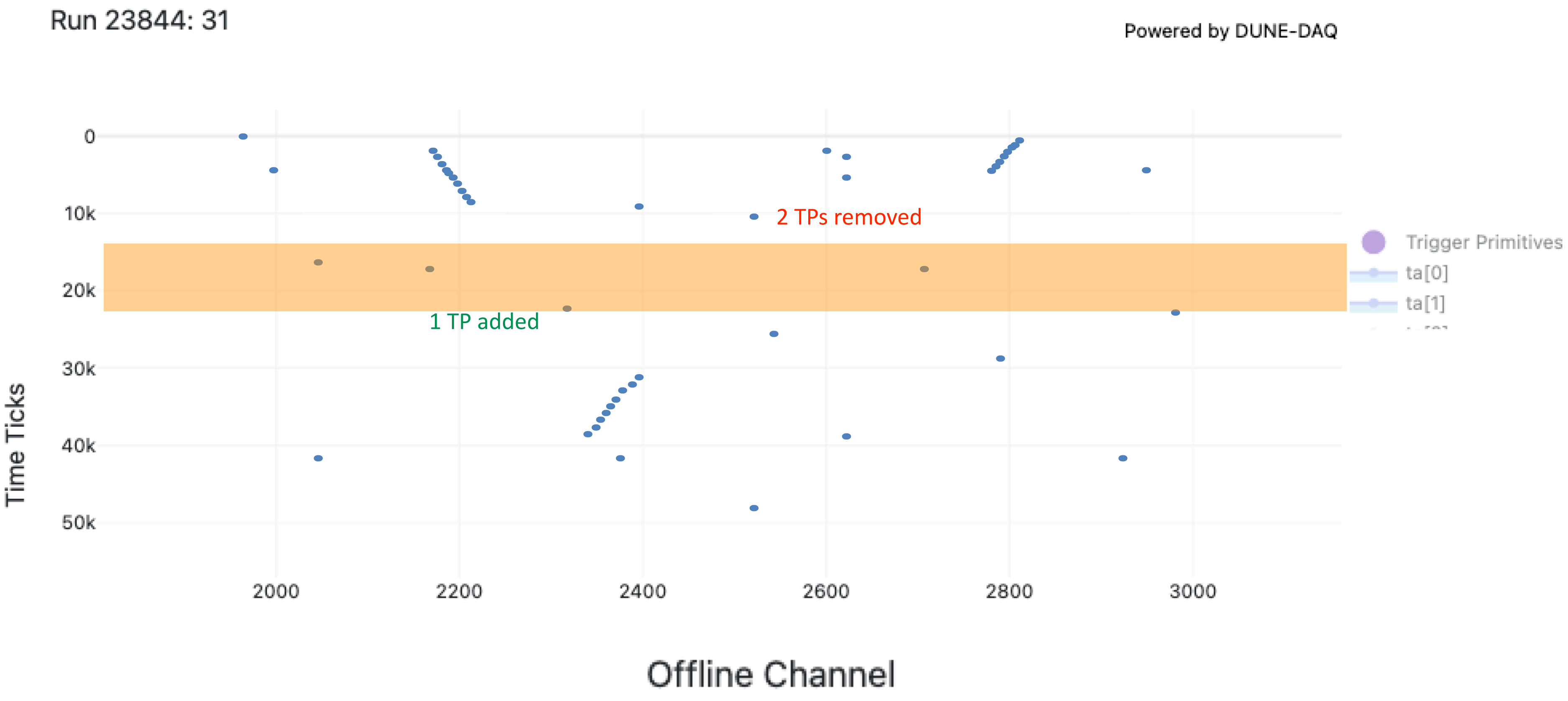
- New window with new TPs
- Check for activities (eg, adjacency_threshold = 8; adj_tolerance = 2)
- False in this case → move the window by one TP only (next slide)



For illustration purpose!

HMTA algorithm

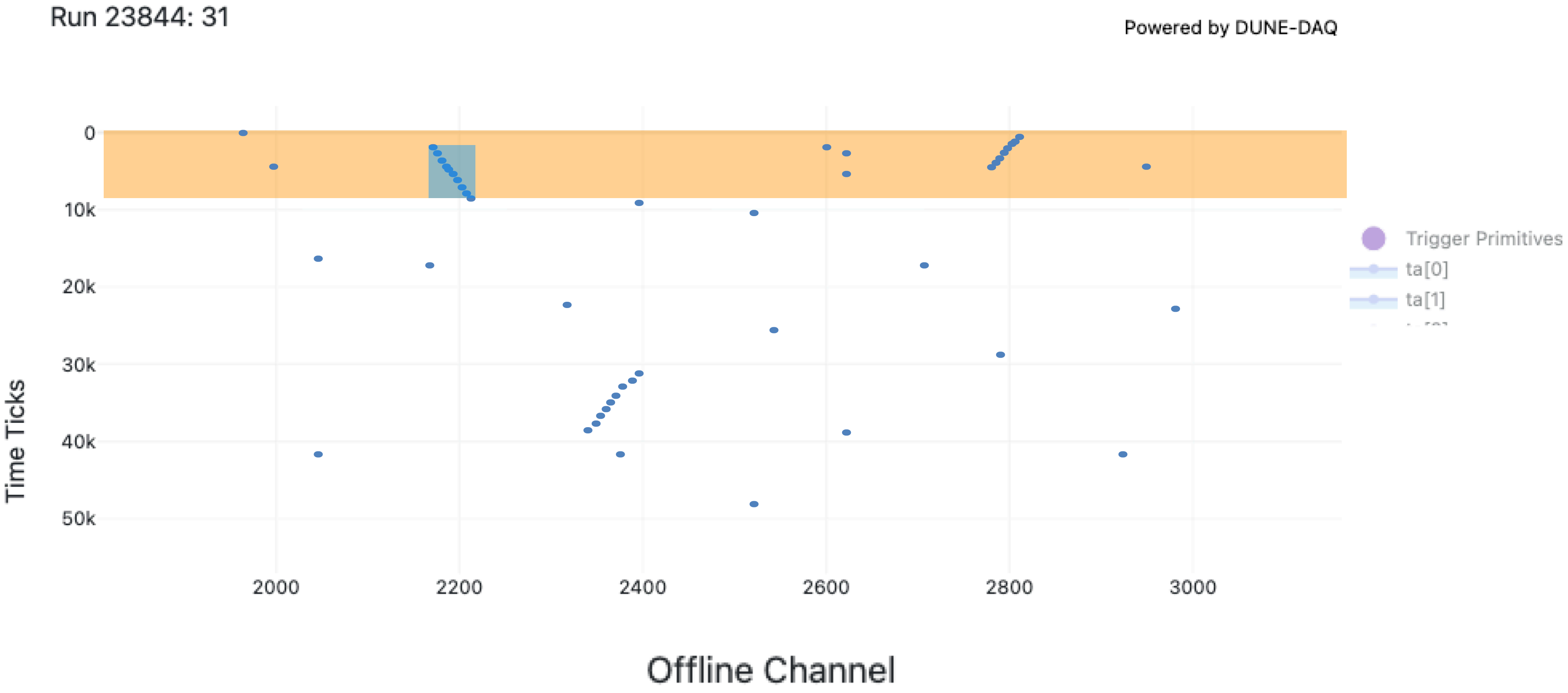
- Fill the new window with new TPs
- Check for activities (eg, adjacency_threshold = 8; adj_tolerance = 2)
- False in this case → move the window by one TP only → Repeat the process



For illustration purpose!

Channel Adjacency TA (CATA) algorithm

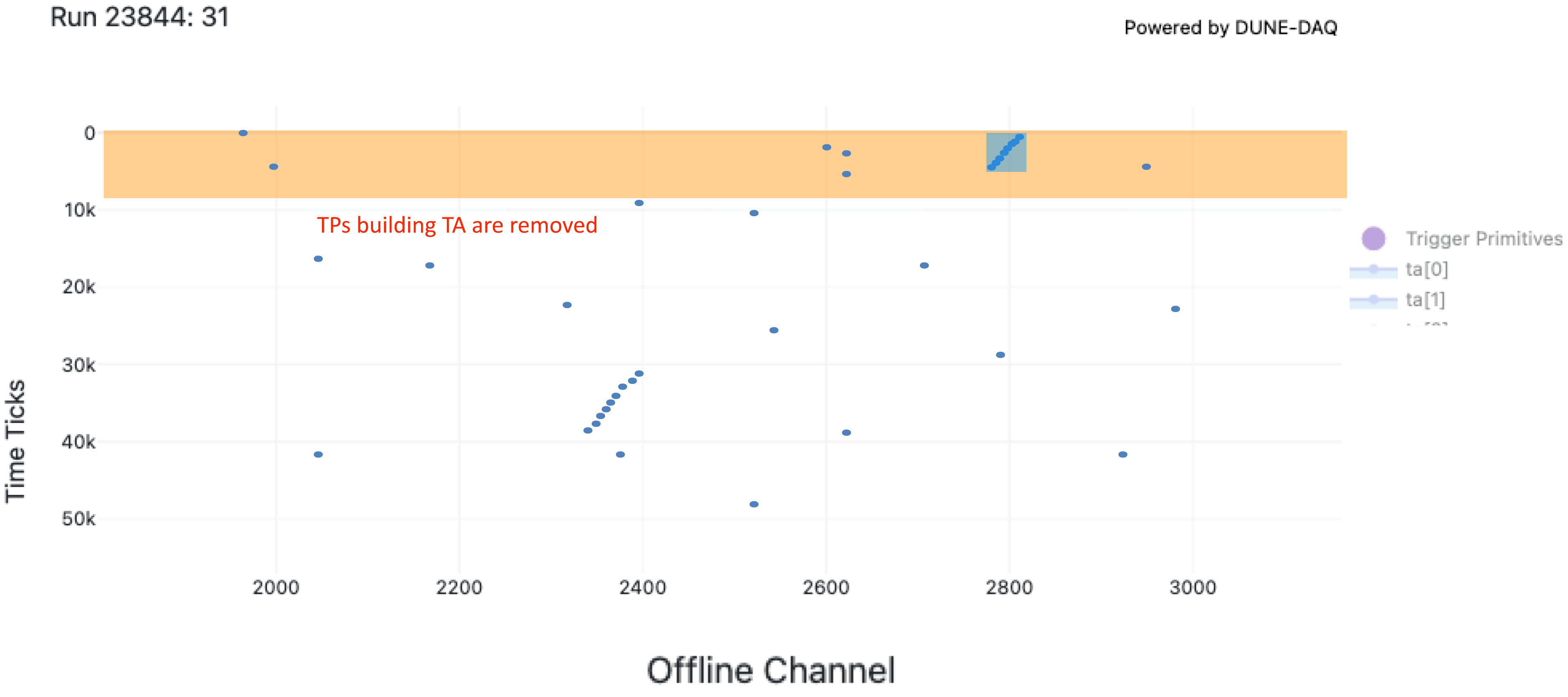
- **Same logic but different TA boundaries**
- **Y-axis:** time_start = time of track's first TP
  time_end = time of track's last TP
- **X-axis:** first_channel = lowest channel among all track TPs
  last_channel = highest channel among all track TPs



For illustration purpose!

Channel Adjacency TA (CATA) algorithm

- **Same logic but different TA boundaries; > 1 TAs allowed per window**
- **Y-axis:** time_start = time of track's first TP
  time_end = time of track's last TP
- **X-axis:** first_channel = lowest channel among all track TPs
  last_channel = highest channel among all track TPs

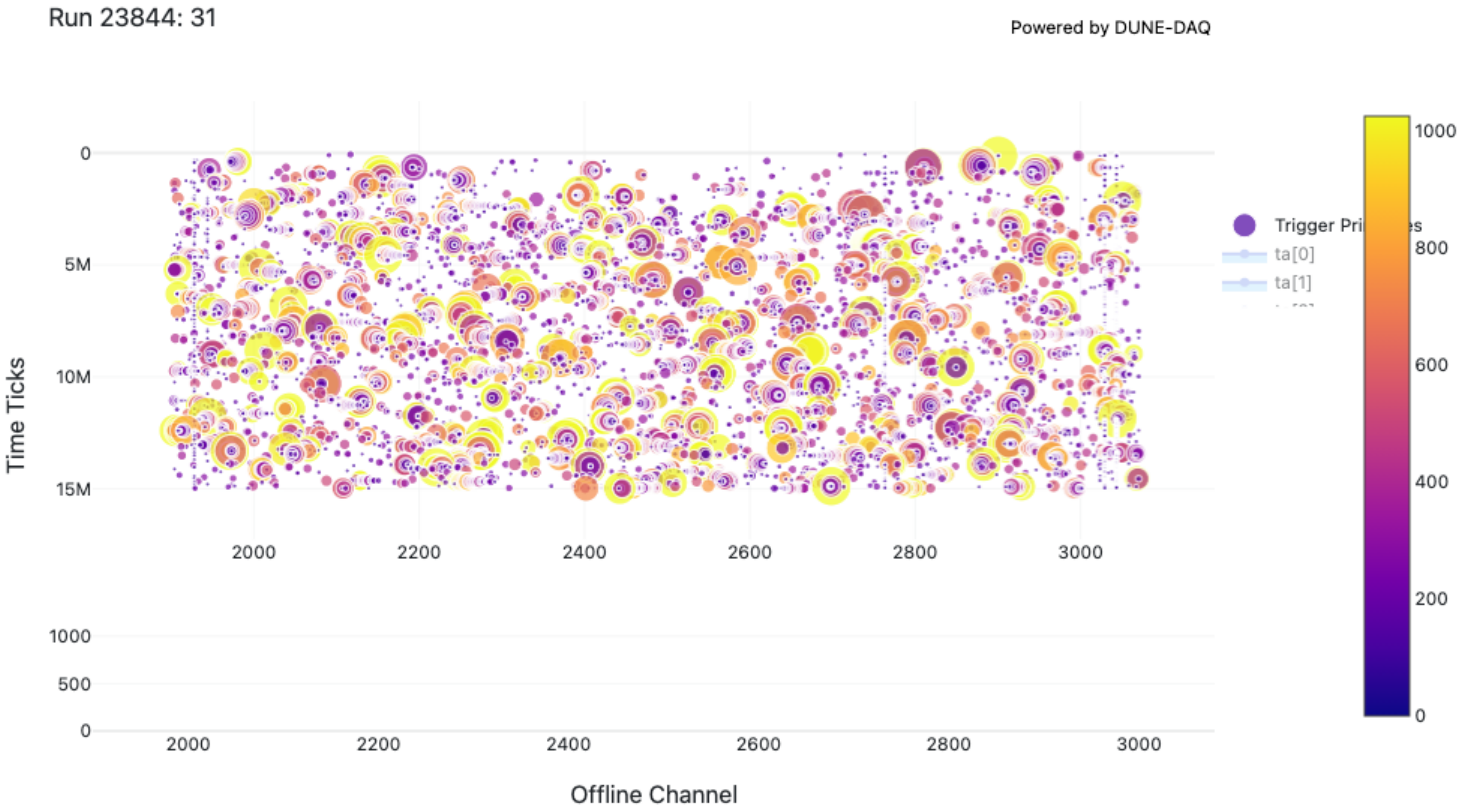


TPs building TA are removed

For illustration purpose!

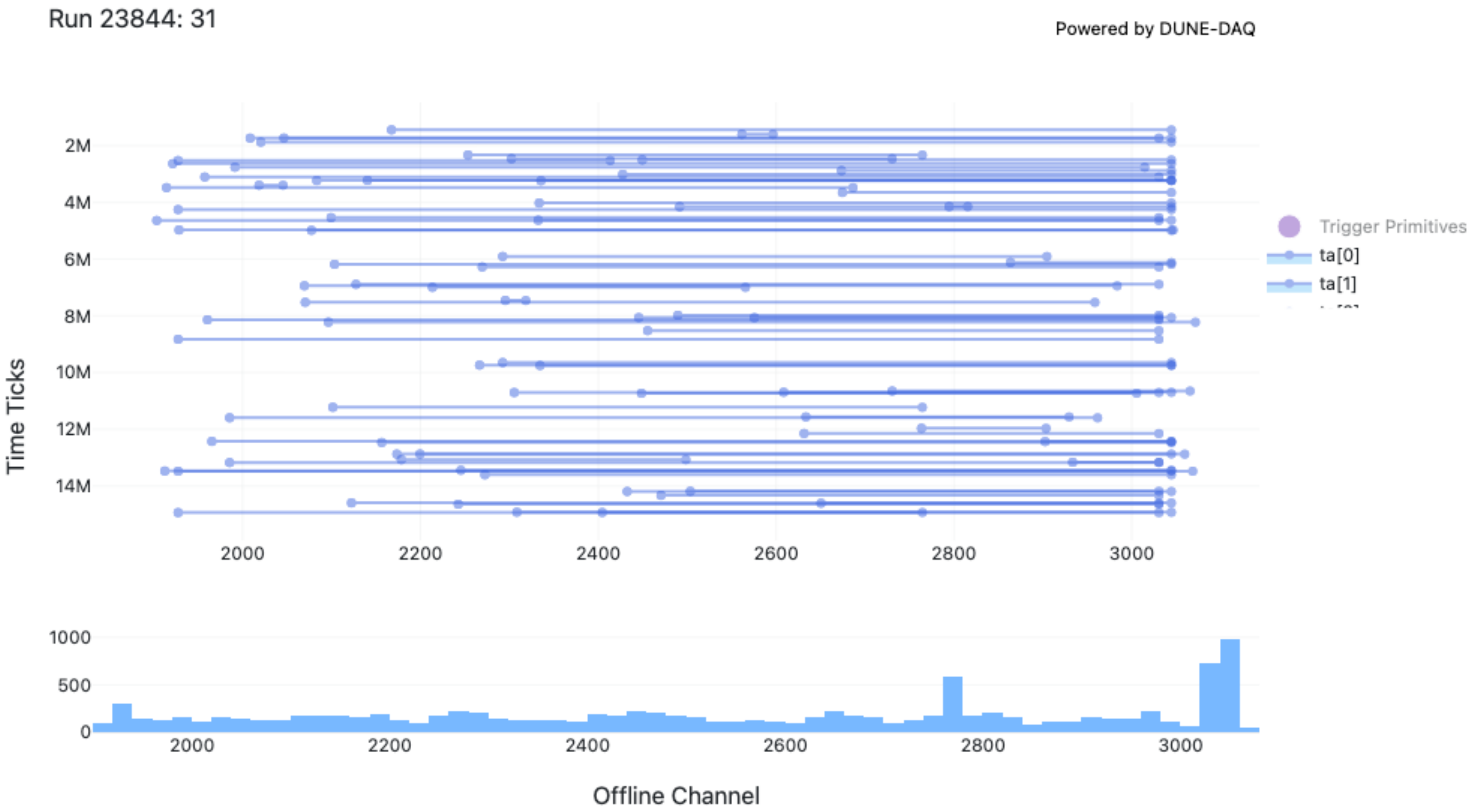
TPs in a timeslice

- Let’s search for activities with adjacency_threshold = 20; adj_tolerance = 5



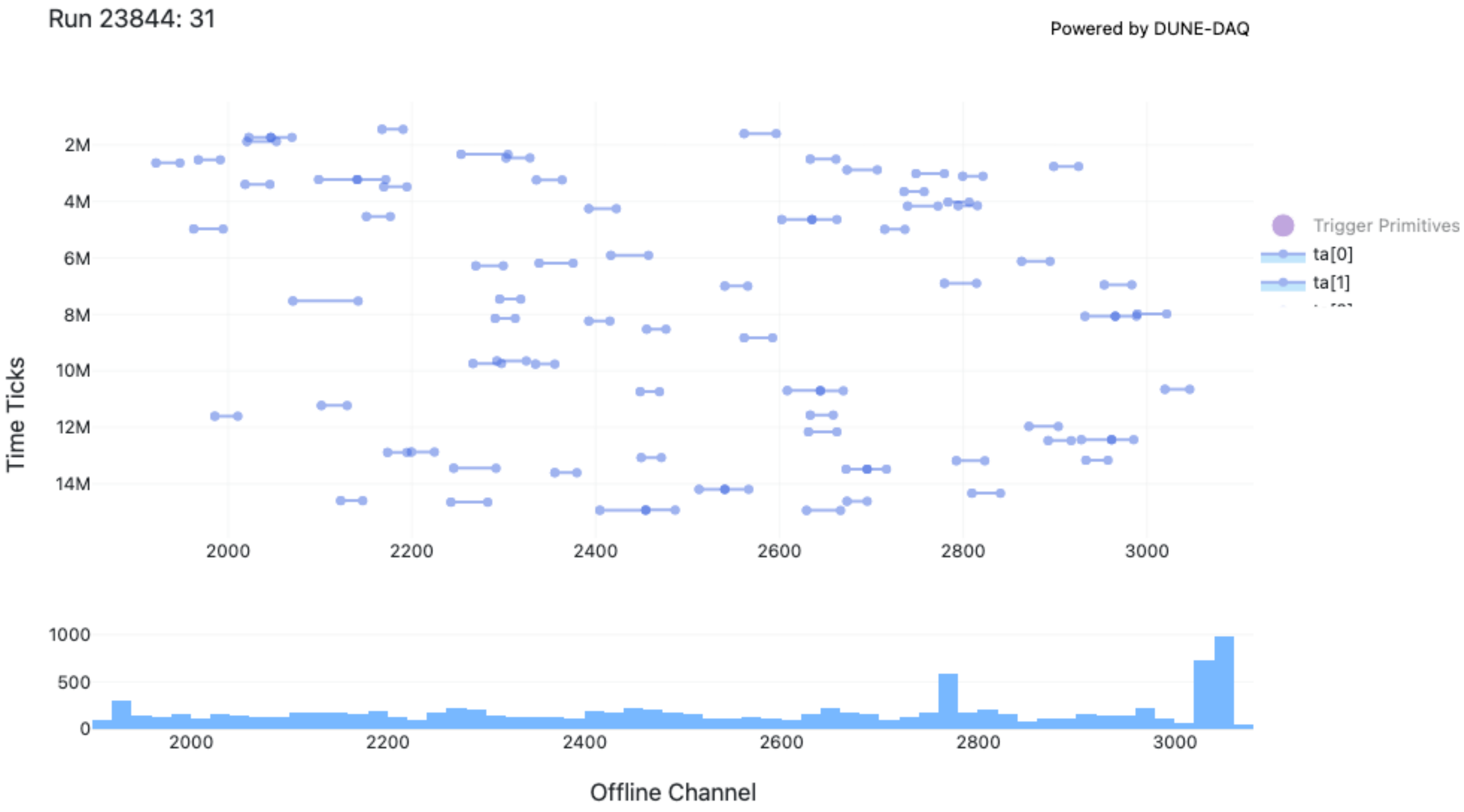
HMTA algorithm

- **Algorithm makes 78 TAs**



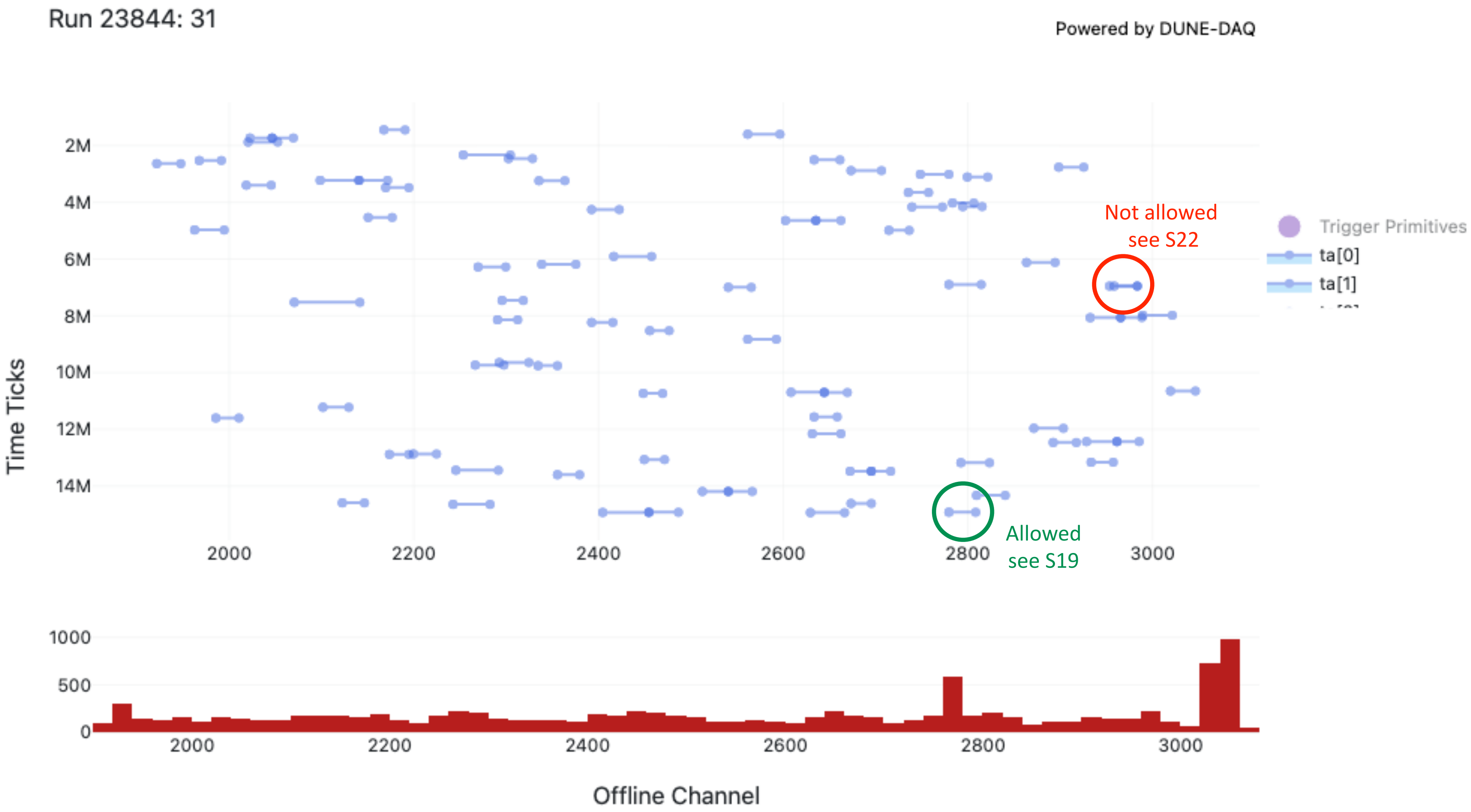
CATA algorithm

- **Algorithm makes same 78 TAs**



CATA algorithm

- **Algorithm makes 79 TAs when > 1 TAs are allowed per window**

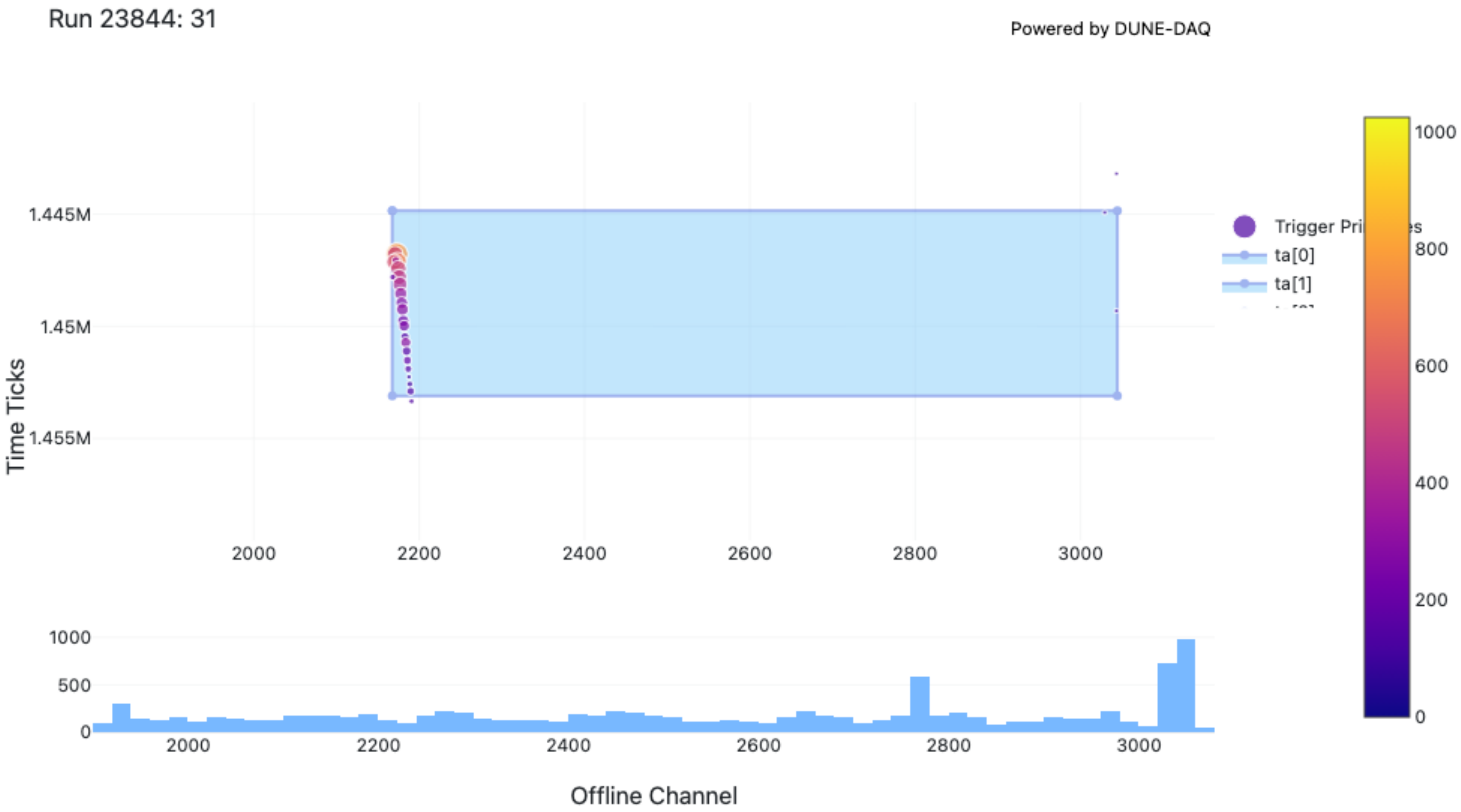


Not allowed see S22

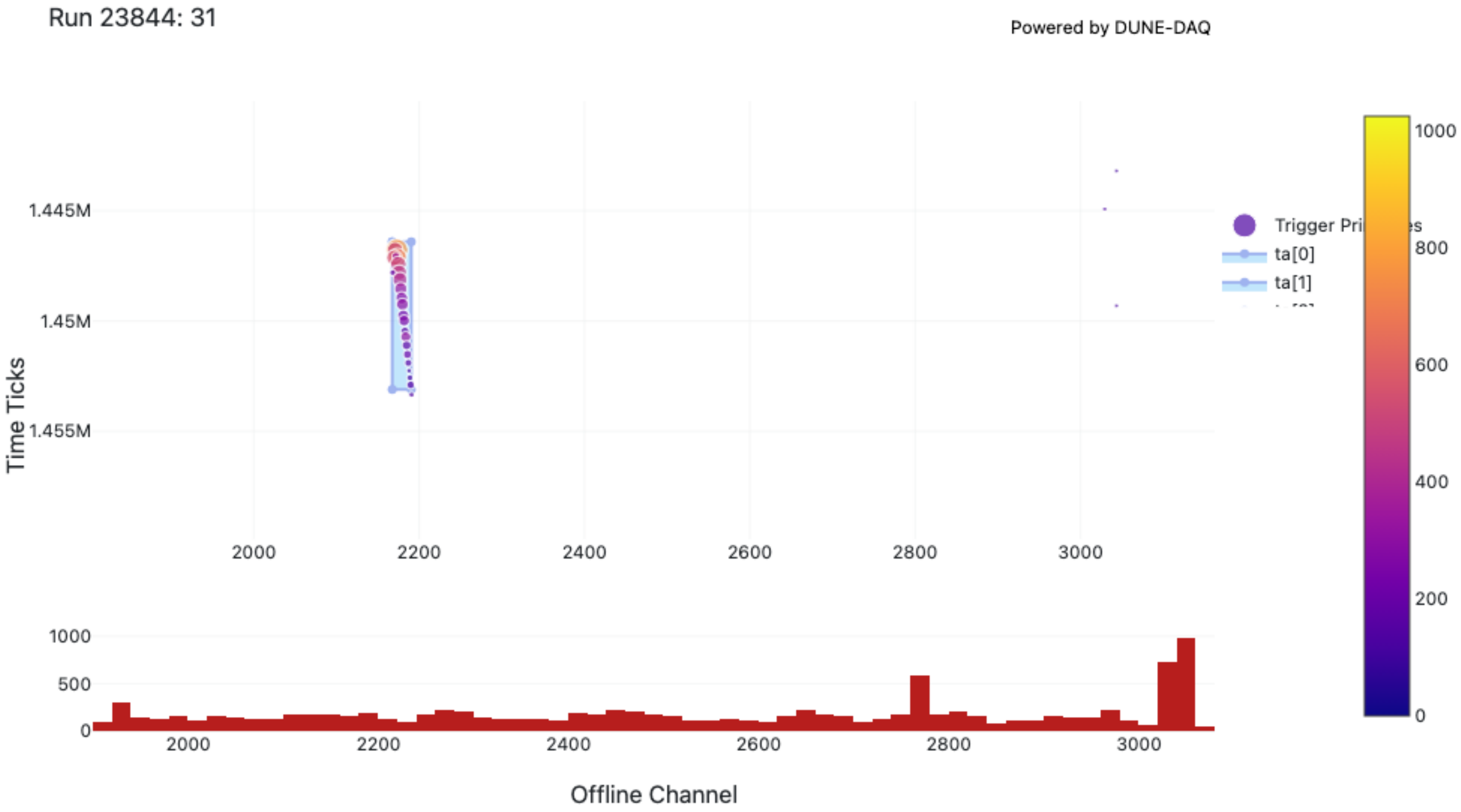
Allowed see S19

HMTA algorithm

- **Window has only one TA**

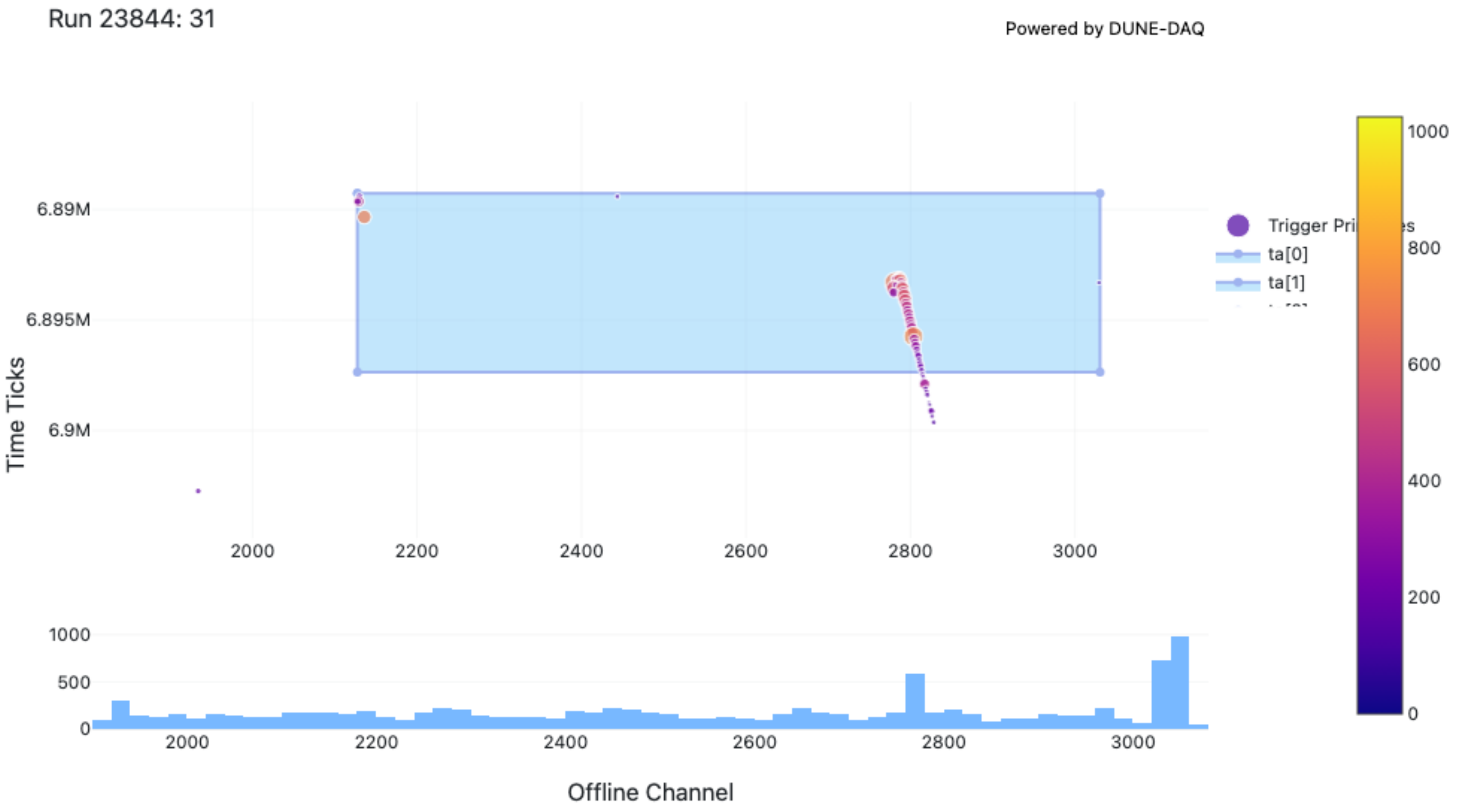


CATA algorithm

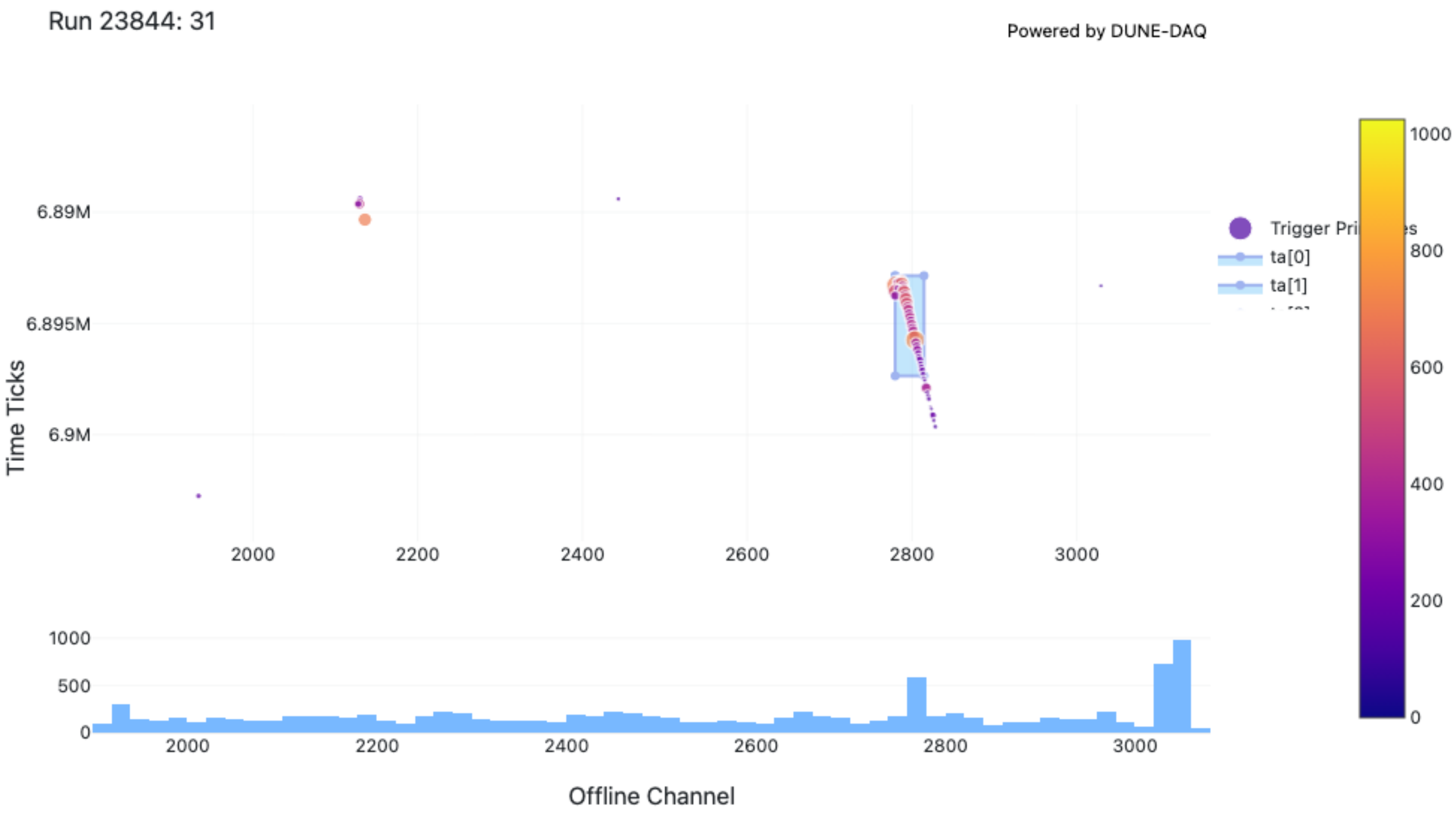


HMTA algorithm

- **Window has only one TA**

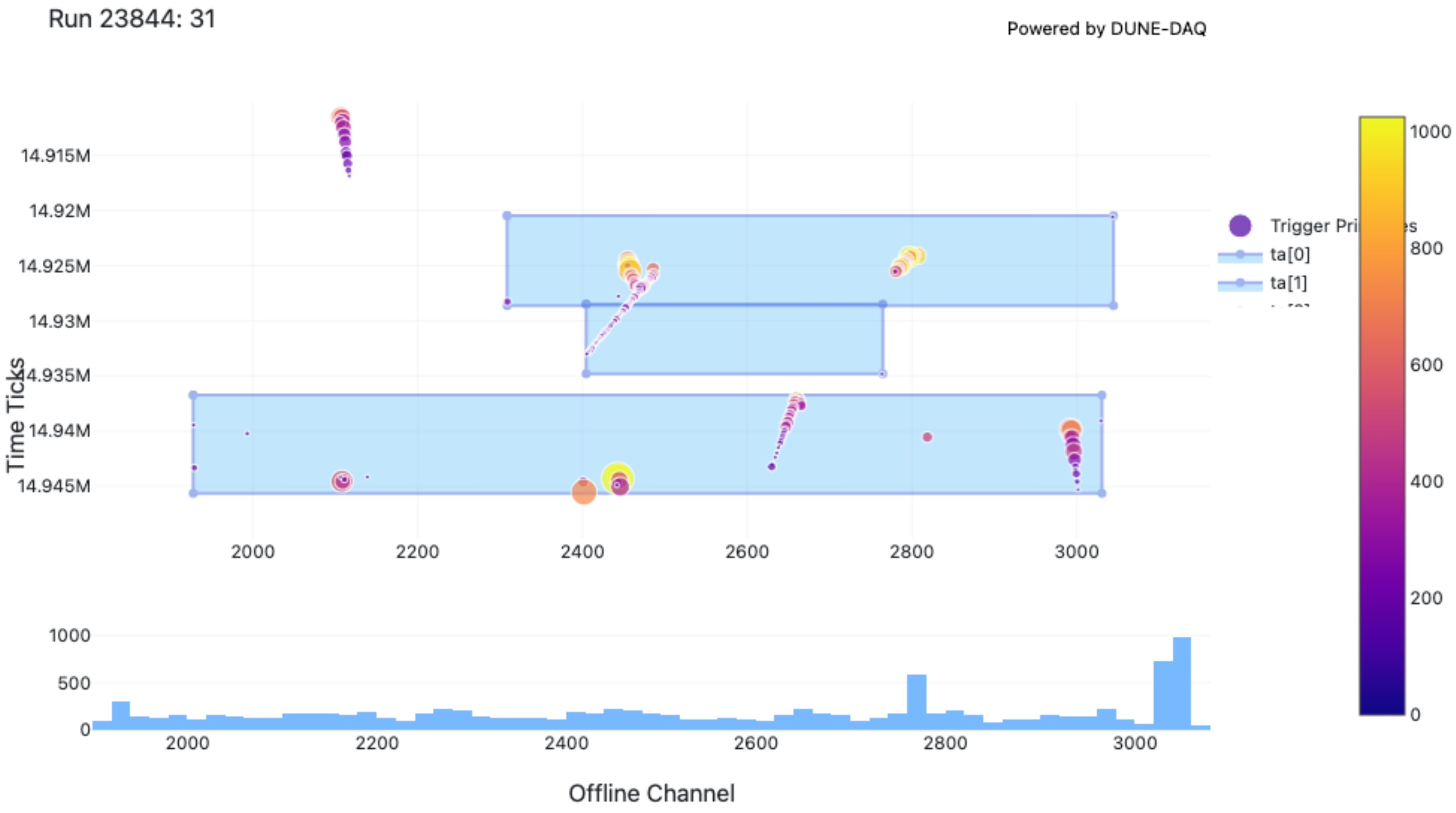


CATA algorithm



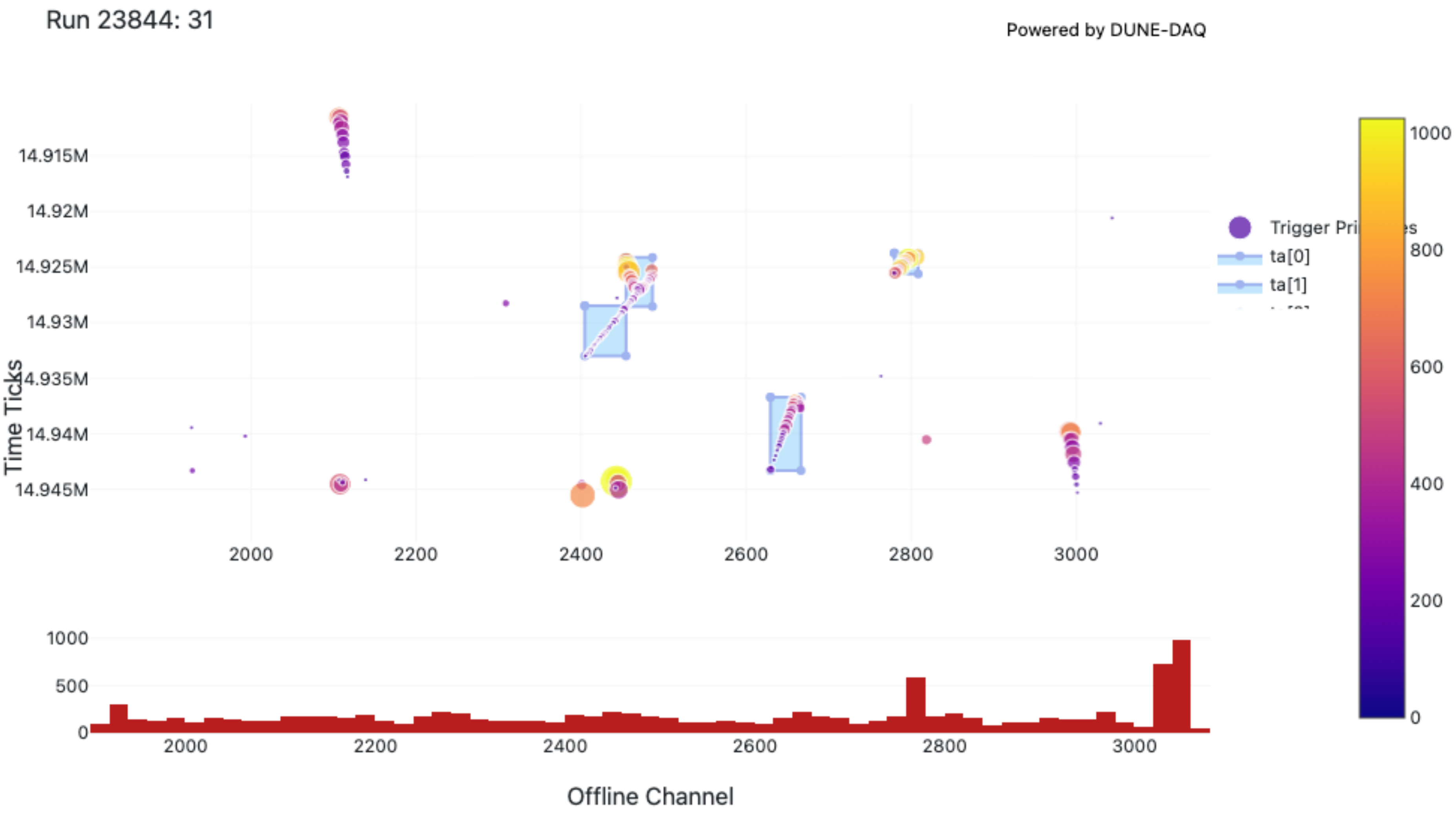
HMTA algorithm

- **Three windows → three TAs**

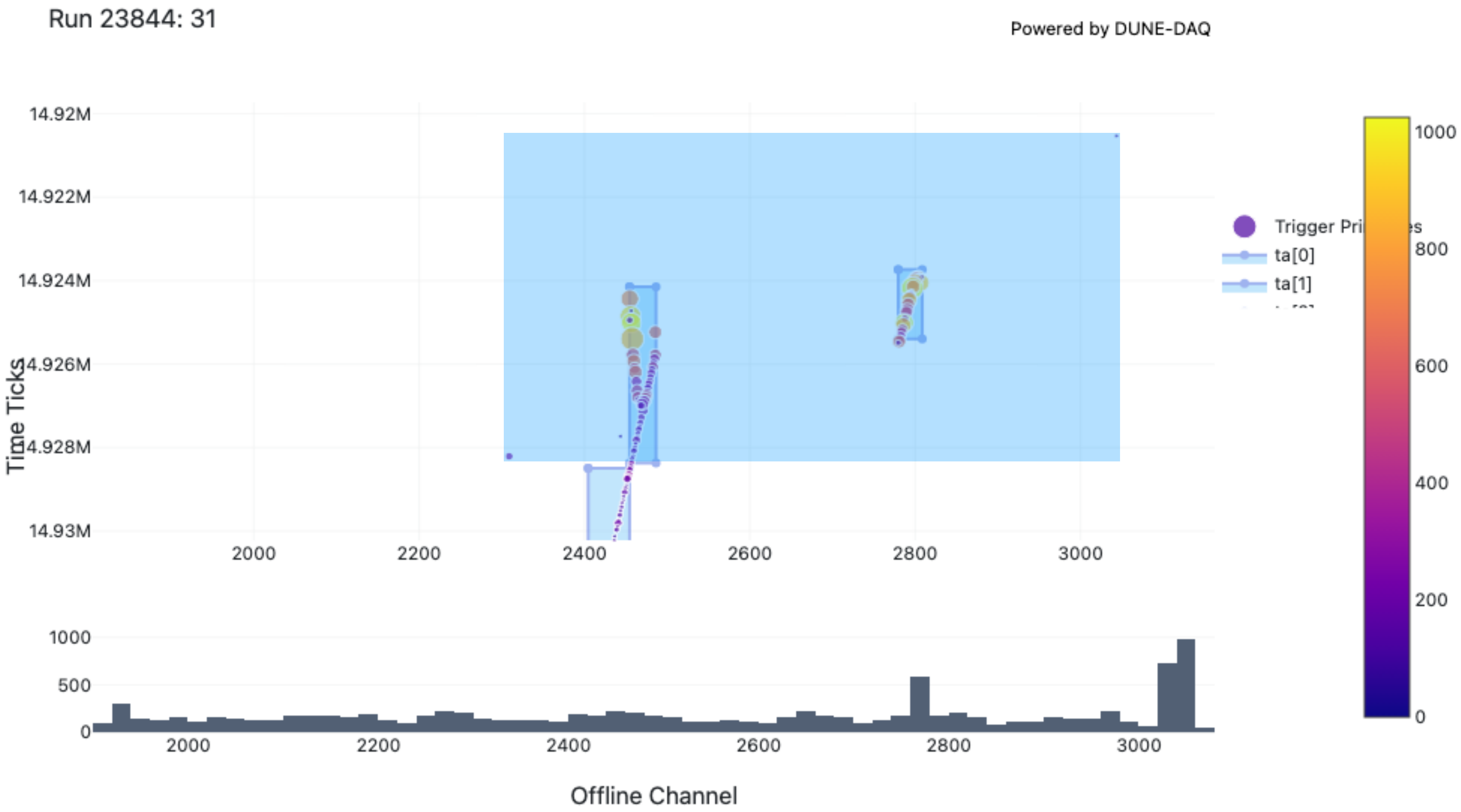


CATA algorithm

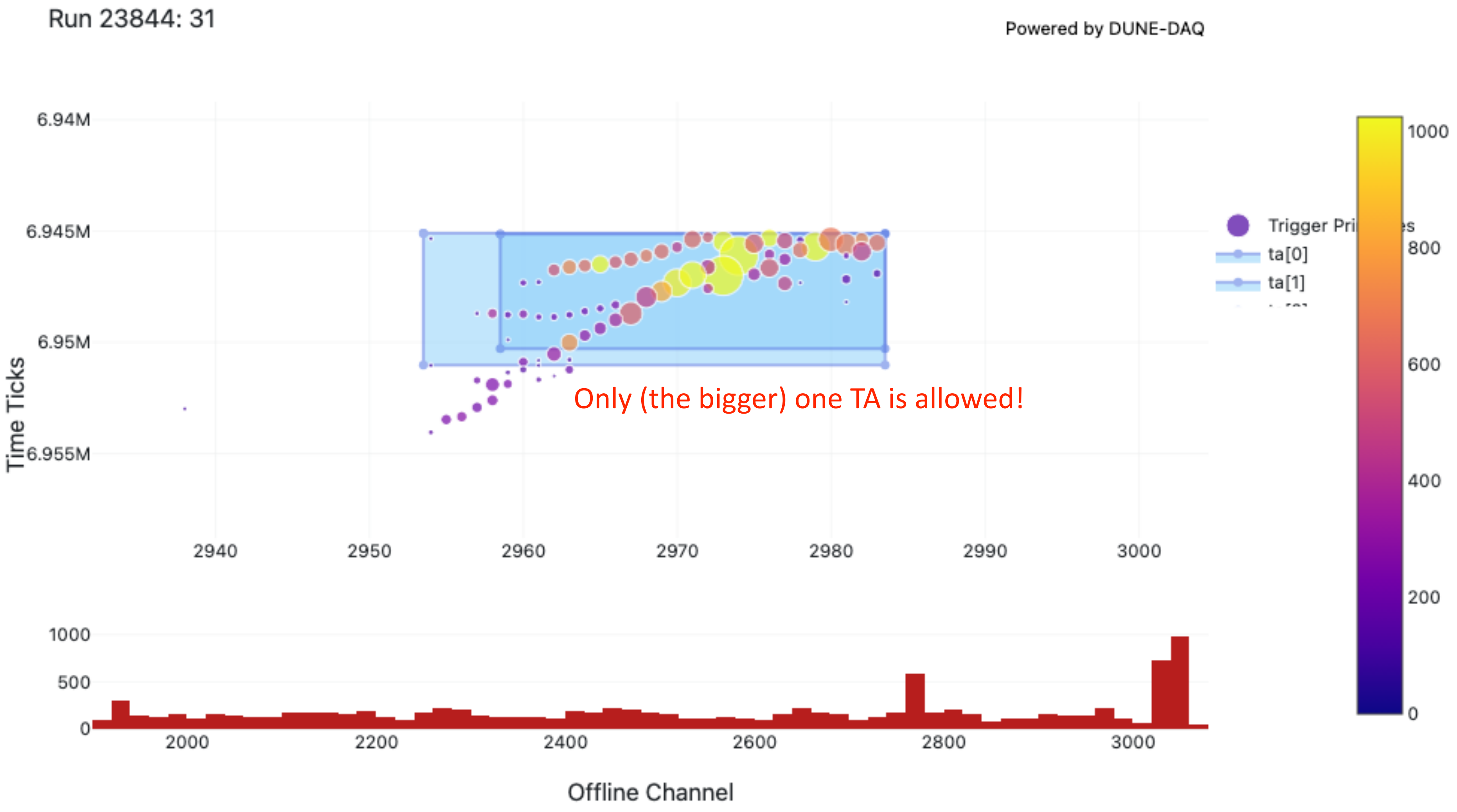
- **Three windows → four TAs**



HMTA and CATA algorithms direct comparison



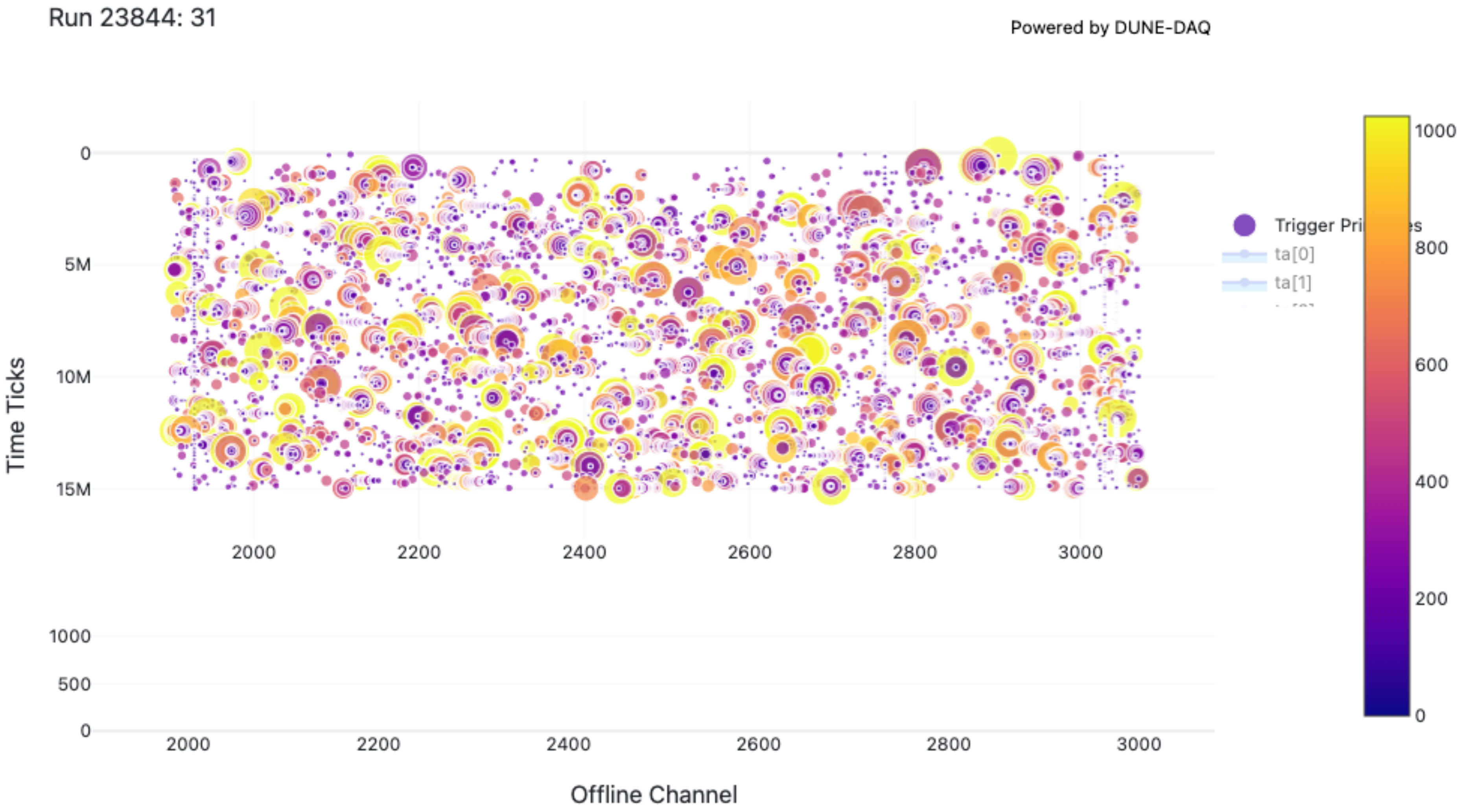
CATA algorithm



Only (the bigger) one TA is allowed!

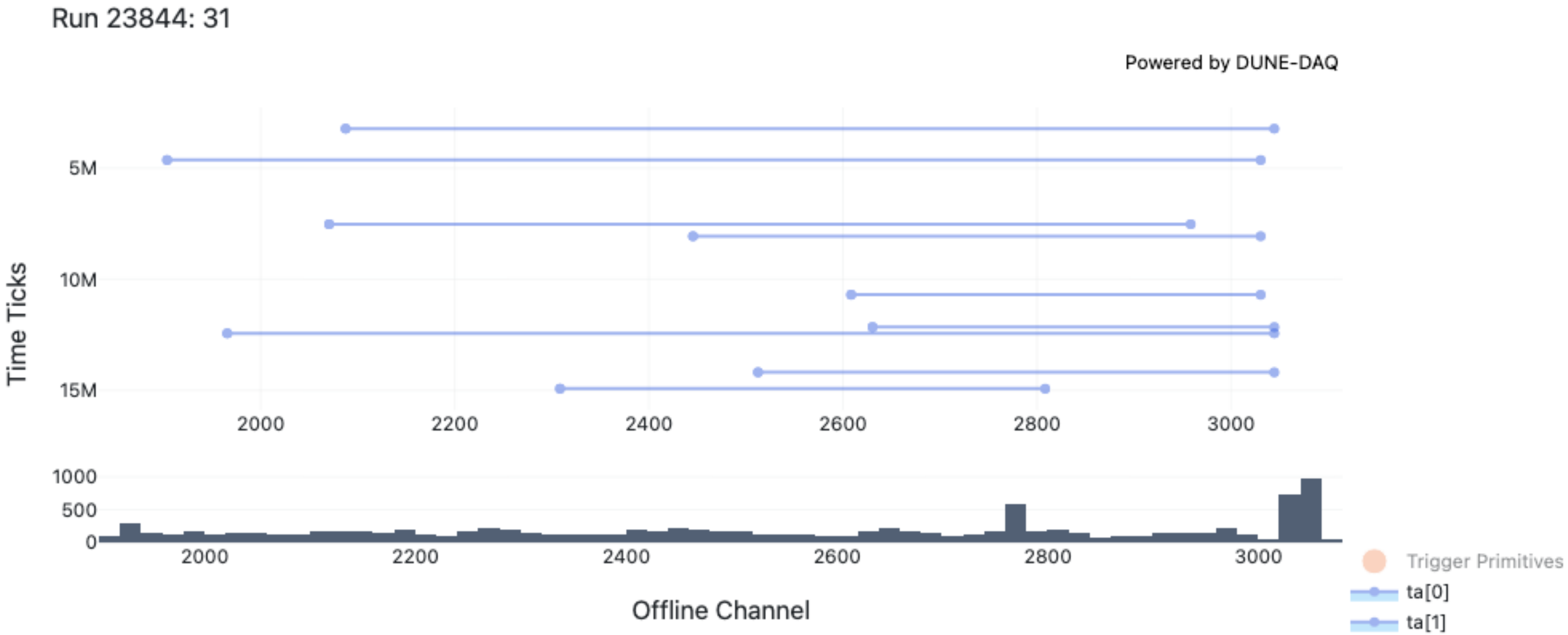
TPs in a timeslice

- Let’s search for activities with adjacency_threshold = 50; adj_tolerance = 5



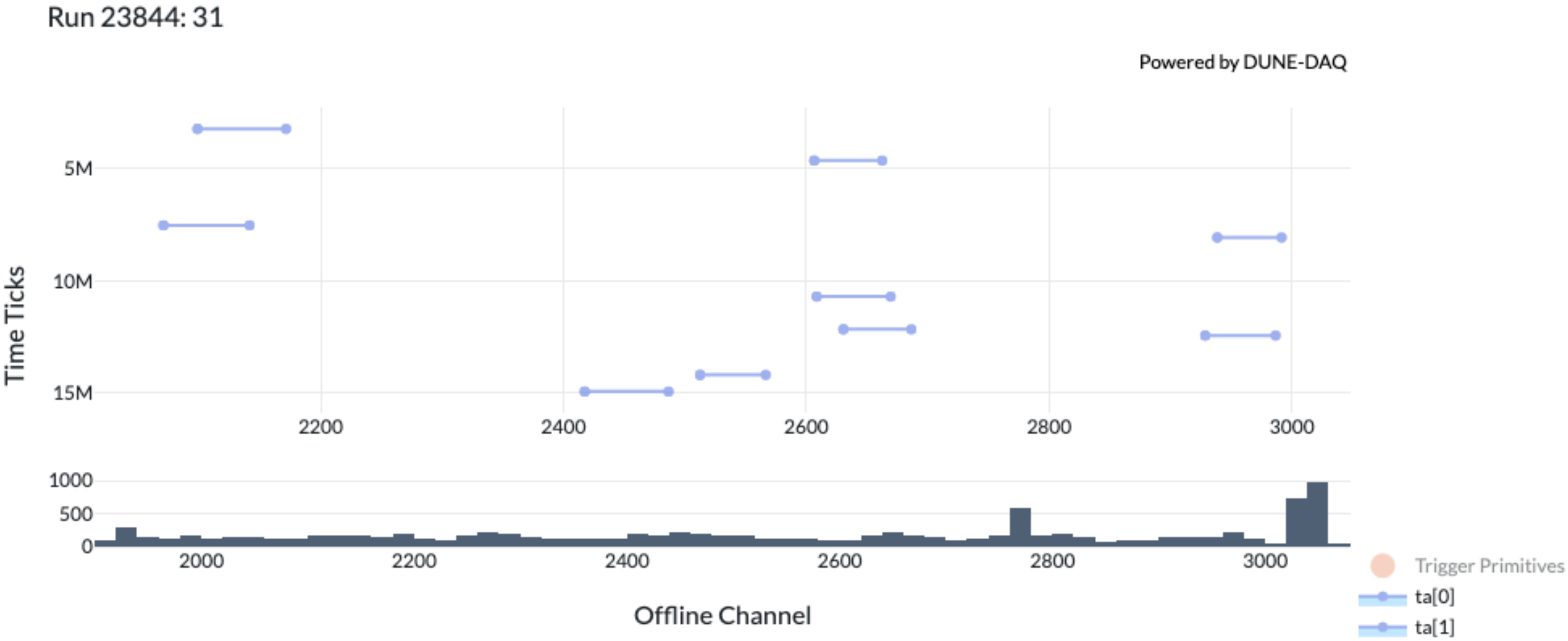
HMTA algorithm

- **Algorithm makes 9 TAs**



CATA algorithm

- **Algorithm makes same 9 TAs (no > 1 TAs per window found!)**



Code changes:

Branch: schhibra/ChannelAdjacency (merged to production/v4)

TAMaker:
triggeralgs/include/triggeralgs/**ChannelAdjacency**/TriggerActivityMaker**ChannelAdjacency.**hpp
triggeralgs/src/TriggerActivityMaker**ChannelAdjacency.**cpp

TCMaker (same as HMTC algorithm for the moment):
triggeralgs/include/triggeralgs/**ChannelAdjacency**/TriggerCandidateMaker**ChannelAdjacency.**hpp
triggeralgs/src/TriggerCandidateMaker**ChannelAdjacency.**cpp

triggeralgs/CMakeLists.txt
Triggeralgs/include/triggeralgs/**ChannelAdjacency**/README.md

trgdataformats/include/trgdataformats/TriggerActivityData.hpp
trgdataformats/include/trgdataformats/TriggerCandidateData.hpp

trgdataformats/pybindsrc/trigger_activity.cpp
trgdataformats/pybindsrc/trigger_candidate.cpp

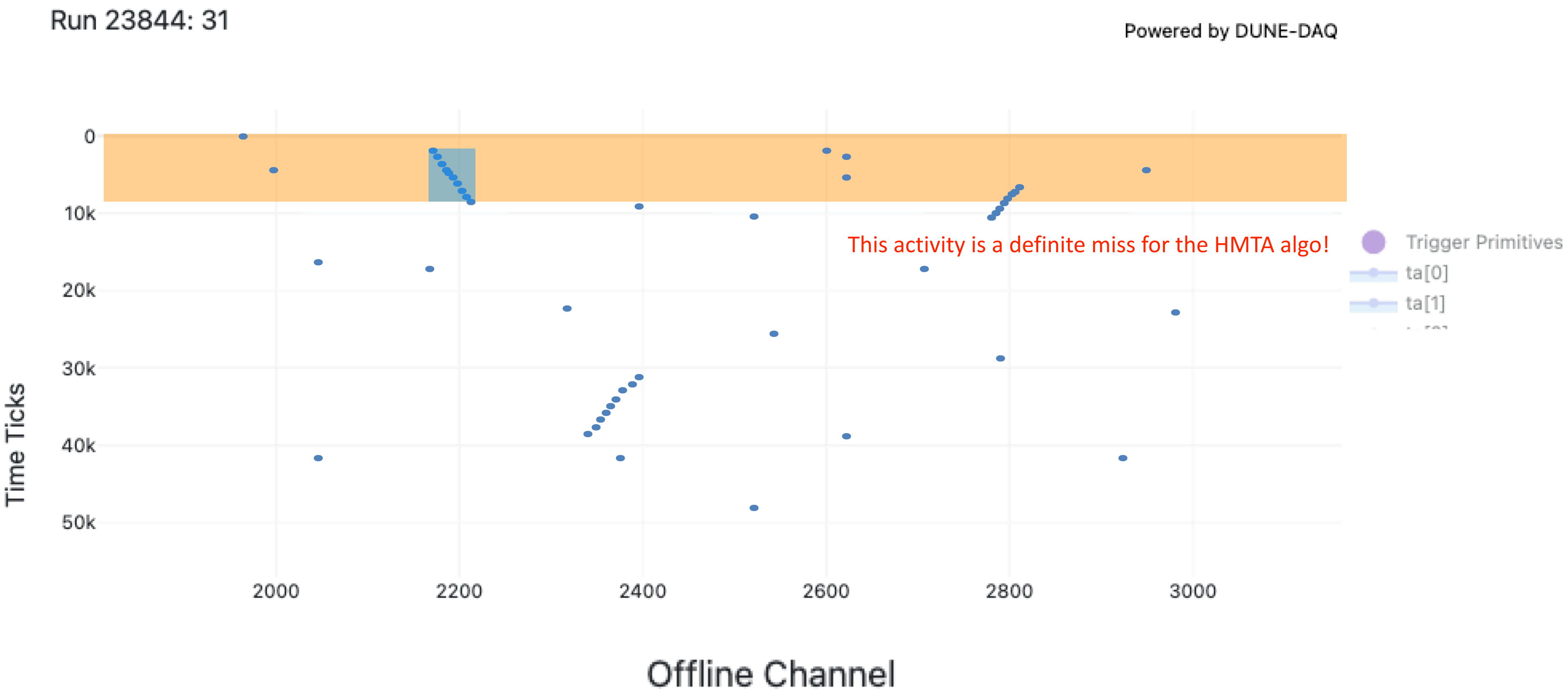
CATA algorithm pros and cons

- **Pros:**
  - TAs' boundaries are better defined now as they should be
    - TAs consume lesser memory → TRs consume lesser memory
    - Lesser raw data to be stored

- **Cons:**
    - overlap of > 1 TAs in time causes duplication of raw data
      - this is a rare case occurs mainly if adjacency_threshold is set to a small number

CATA ongoing work

- eg, adjacency_threshold = 8; adj_tolerance = 2
- Activity found → build a TA
- Remove TPs (next slide)

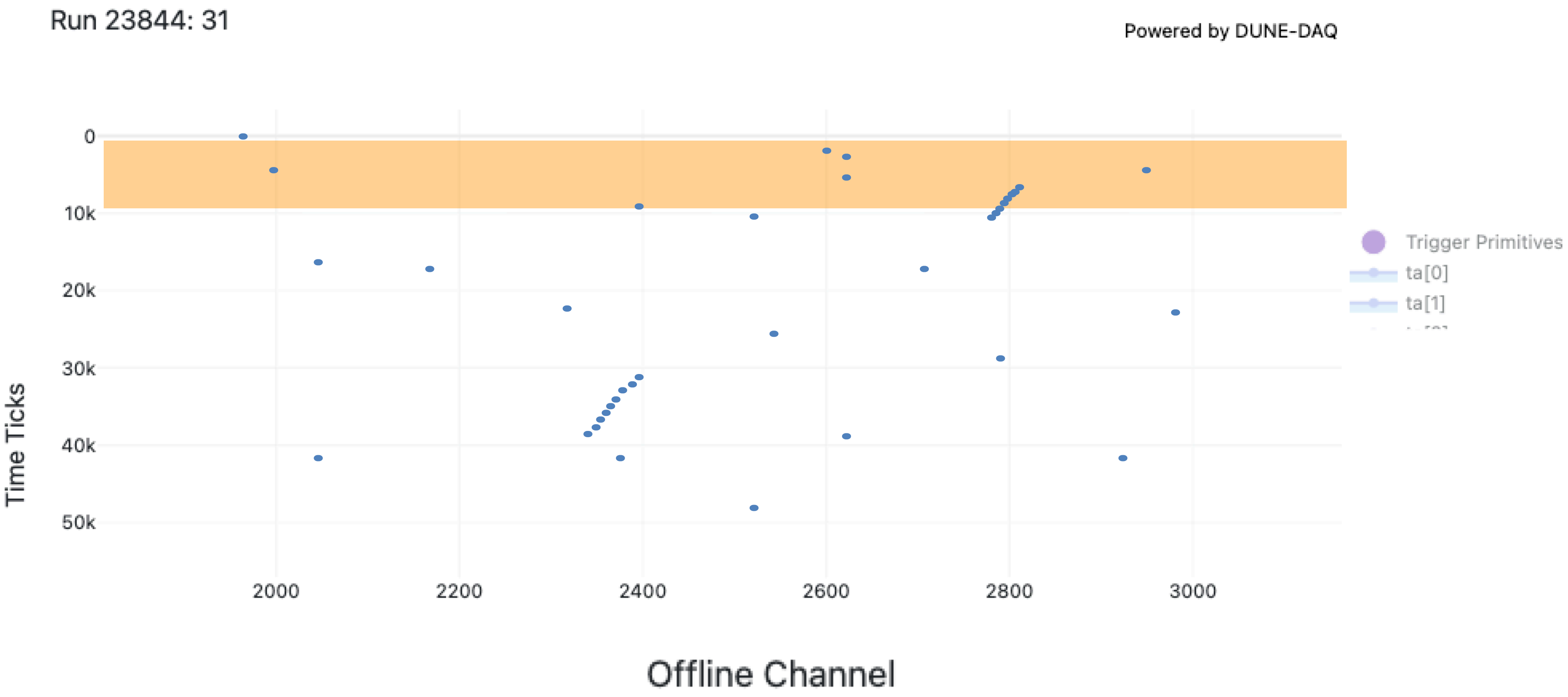


This activity is a definite miss for the HMTA algo!

For illustration purpose!

CATA ongoing work

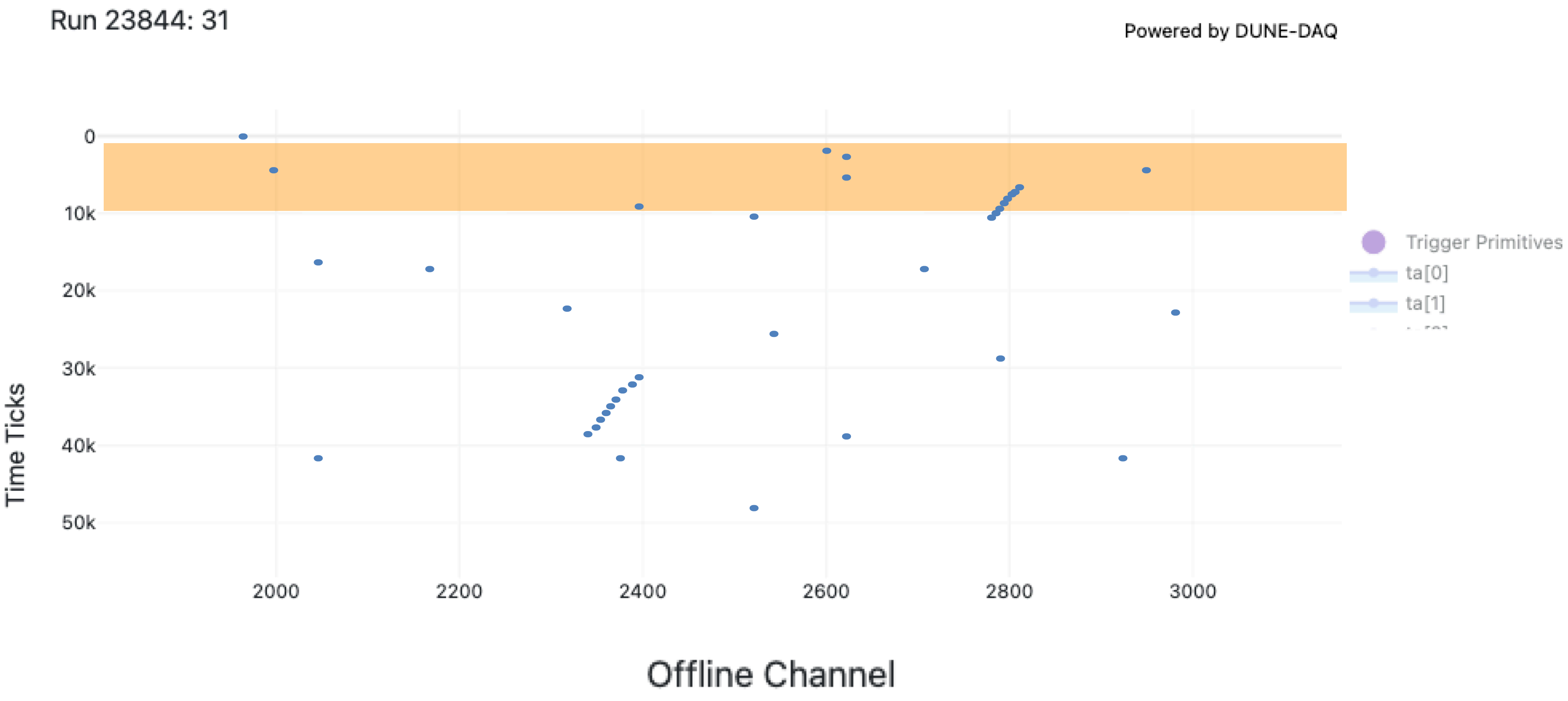
- Search for activity
- No activity found → move the window by one TP only (next slide)



For illustration purpose!

CATA ongoing work

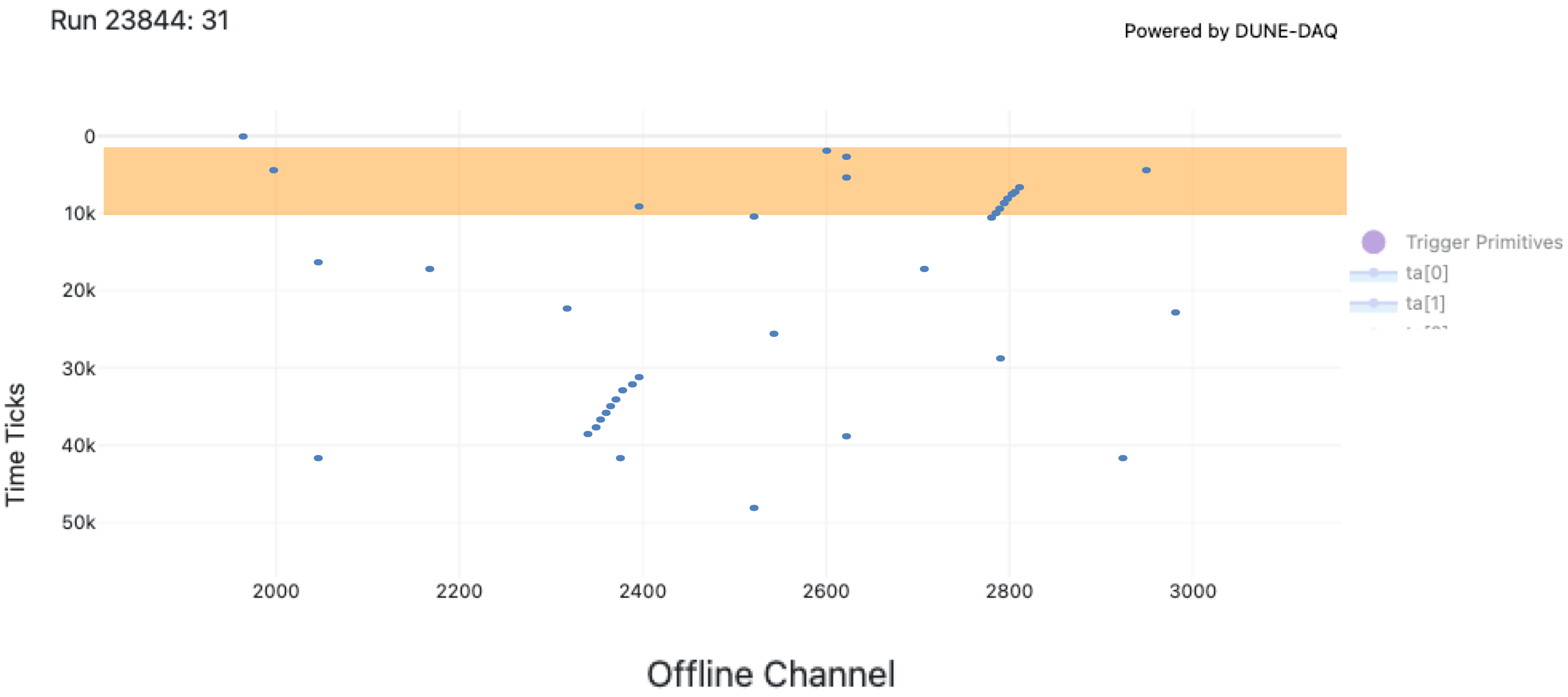
- Search for activity
- No activity found → move the window by one TP only (next slide)



For illustration purpose!

CATA ongoing work

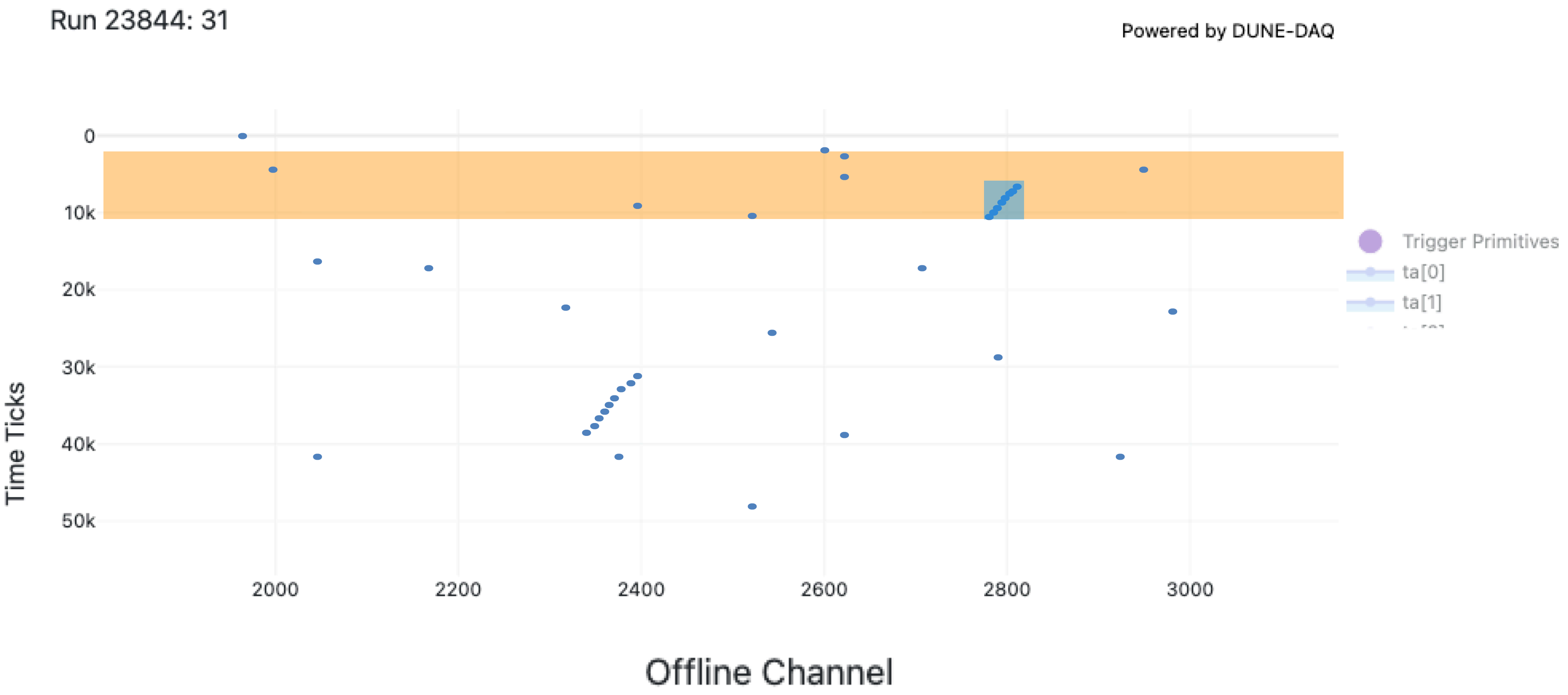
- Search for activity
- No activity found → move the window by one TP only (next slide)



For illustration purpose!

CATA ongoing work

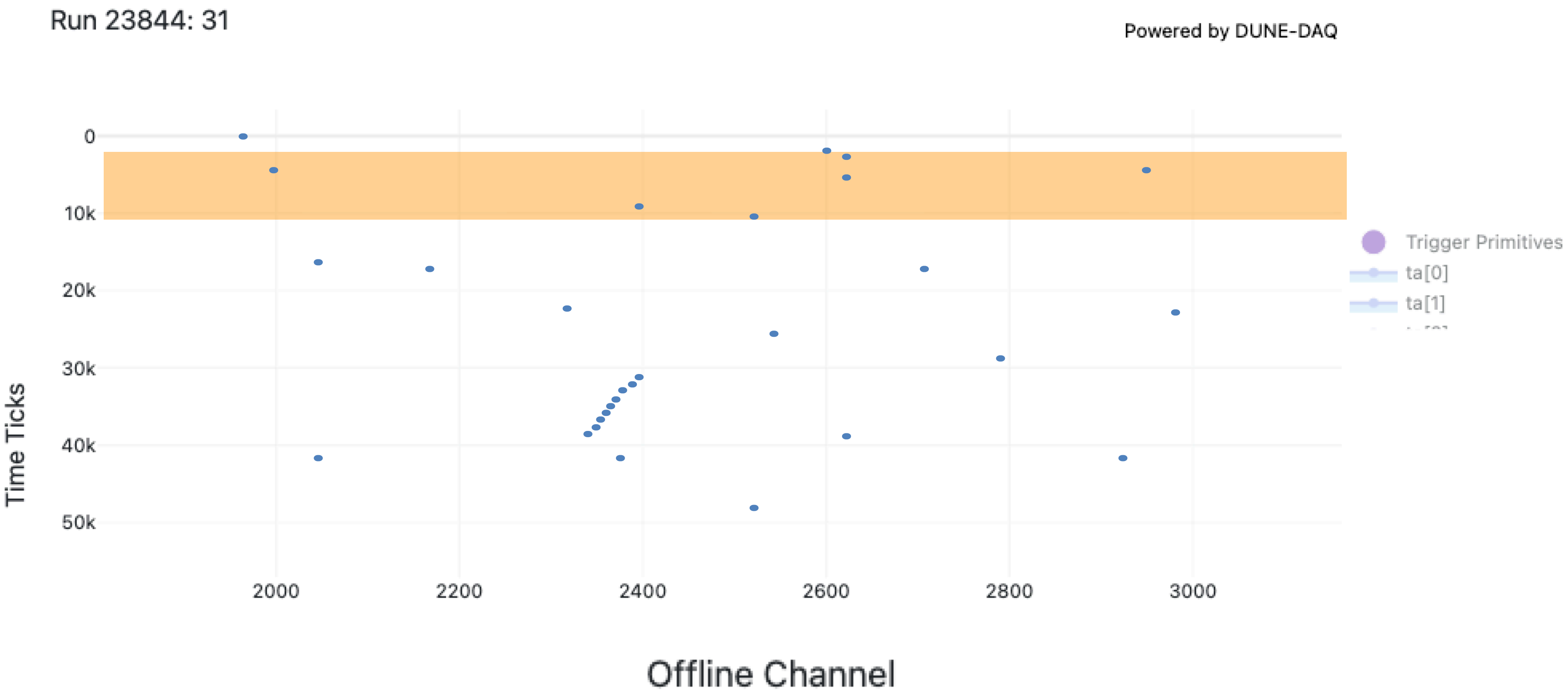
- Search for activity
- Activity found → build a TA
- Remove TPs (next slide)



For illustration purpose!

CATA ongoing work

- Search for activity
- No activity found → move the window by one TP only, and so on



For illustration purpose!

CATA algorithm pros and cons

- **Pros:**
  - TAs' boundaries are better defined now as they should be
    - TAs consume lesser memory → TRs consume lesser memory
    - Lesser raw data to be stored
  - **Capability of building additional TAs (missed by HMTA algorithm)**

- **Cons:**
    - overlap of > 1 TAs in time causes duplication of raw data
      - this is a rare case occurs mainly if adjacency_threshold is set to a small number