LEO OBJECT DETECTION USING BLIND STACKING

Ben Cooke

University of Warwick

NAM 2022

WARWICK
THE UNIVERSITY OF WARWICK

[dstl]
The Science Inside

UKRI

Science and Technology Facilities Council

NAM 2022
AT THE UNIVERSITY OF WARWICK

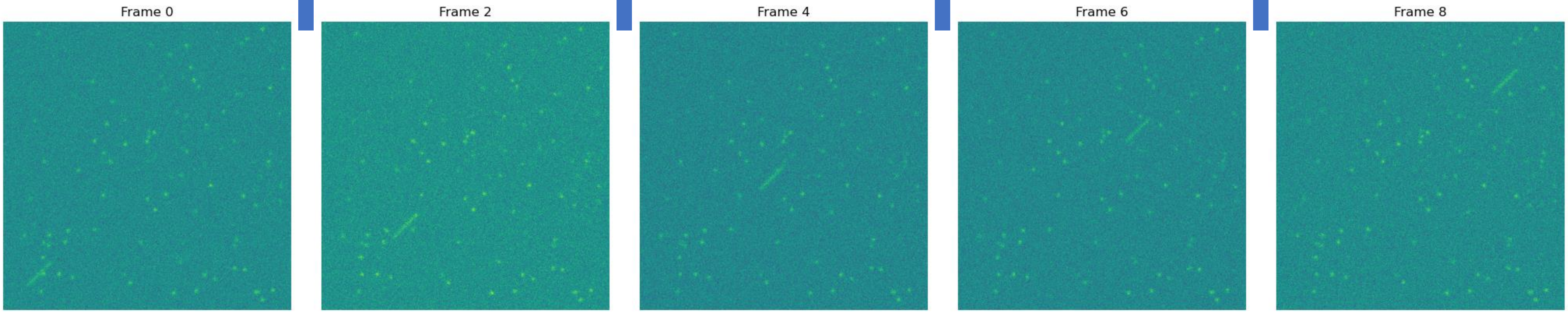
Motivation

- The number of objects in LEO is increasing and will likely continue to do so for the foreseeable future
- Not all of the objects are tracked, especially the smallest and faintest
- Optical observations can provide unique discovery and characterisation opportunities

- How do we detect LEO objects, without requiring prior information on their orbital properties?

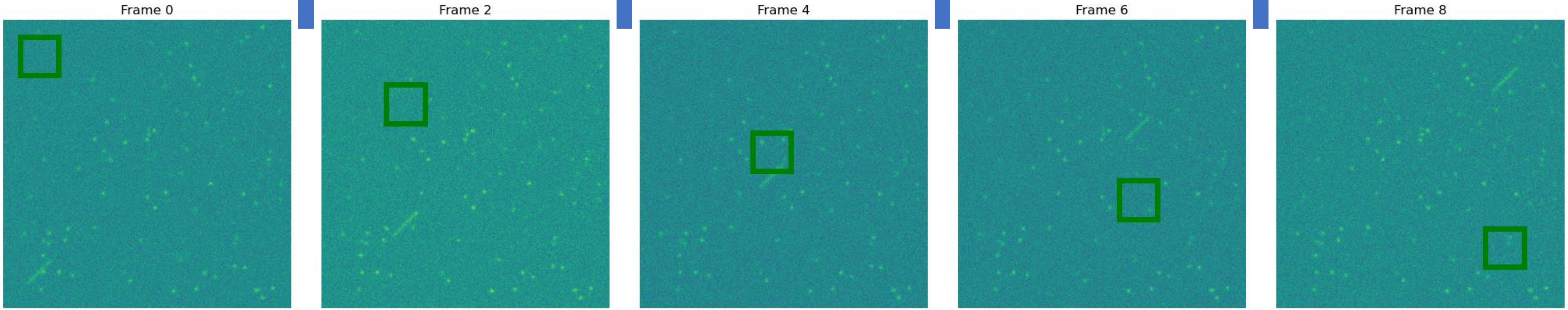
Blind stacking – Introduction

- Method for detection of moving targets with unknown speed, direction and location
- Involves testing many potential paths on a pixel-by-pixel basis
- Procedure:
  1. Choose a path to test (i.e. rate and direction of motion between frames)
  2. Shift each image by the appropriate amount and combine the corresponding pixels (sum/mean/median)
  3. Compare stacked image with a master, replacing brighter pixels (and their meta data)
  4. Repeat steps 1-3 for each potential path



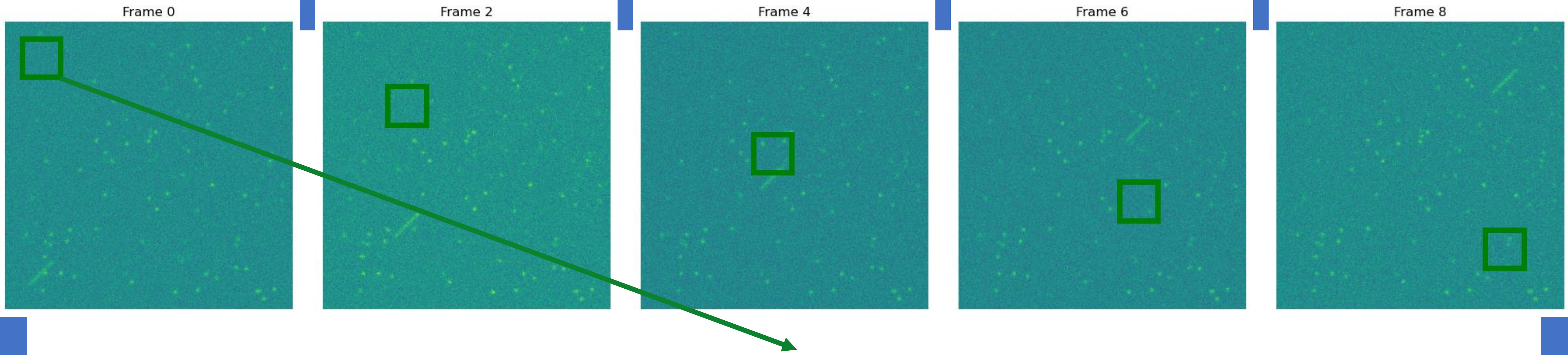
Individual frames containing a target moving (50,50) pixels per frame. Not all frames are shown.

To identify the target, multiple paths must be tested.



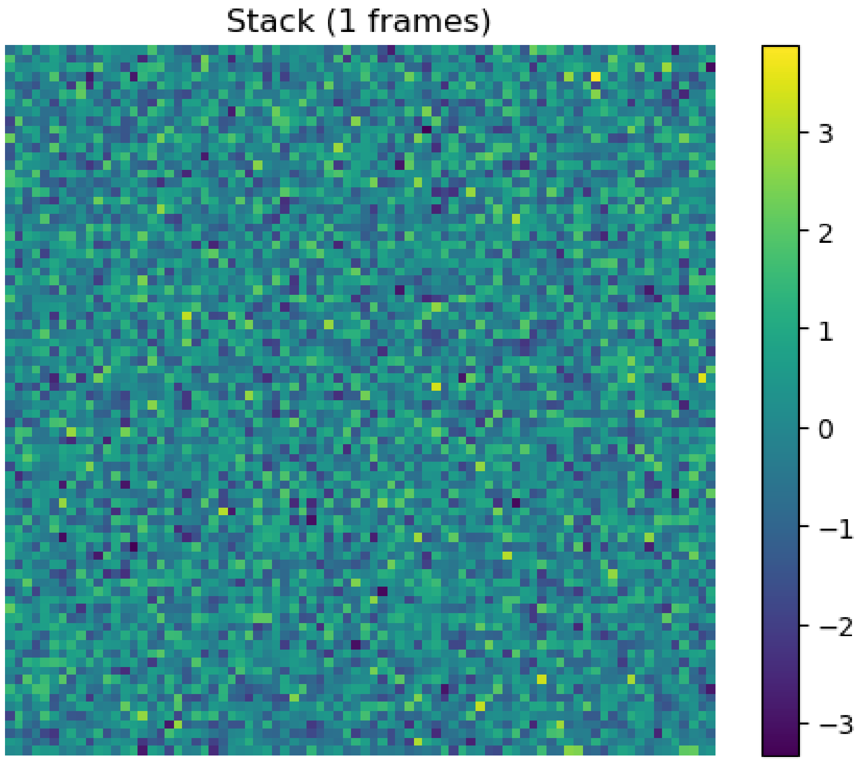
Tested path:
50 pixels South
50 pixels East
(-50,50)

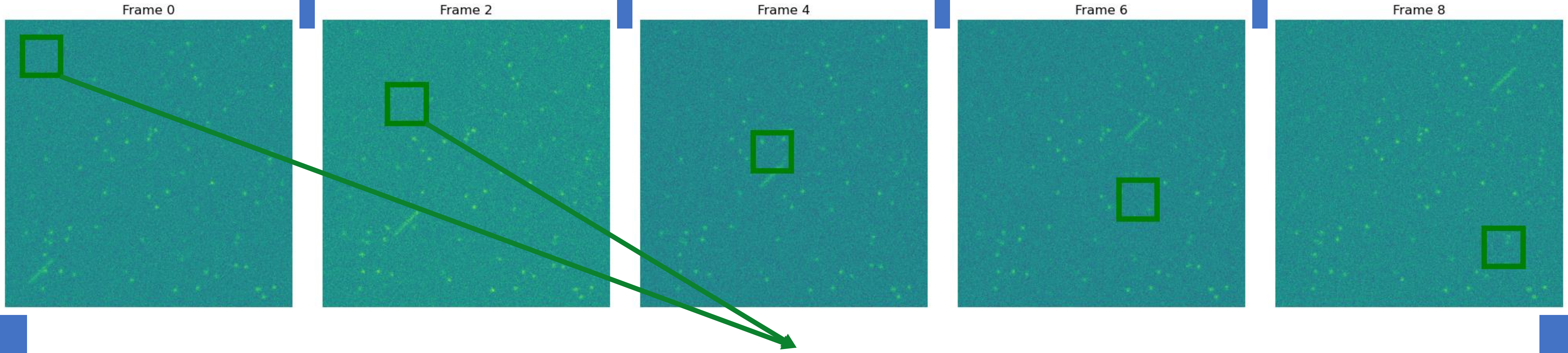
This is an
incorrect path

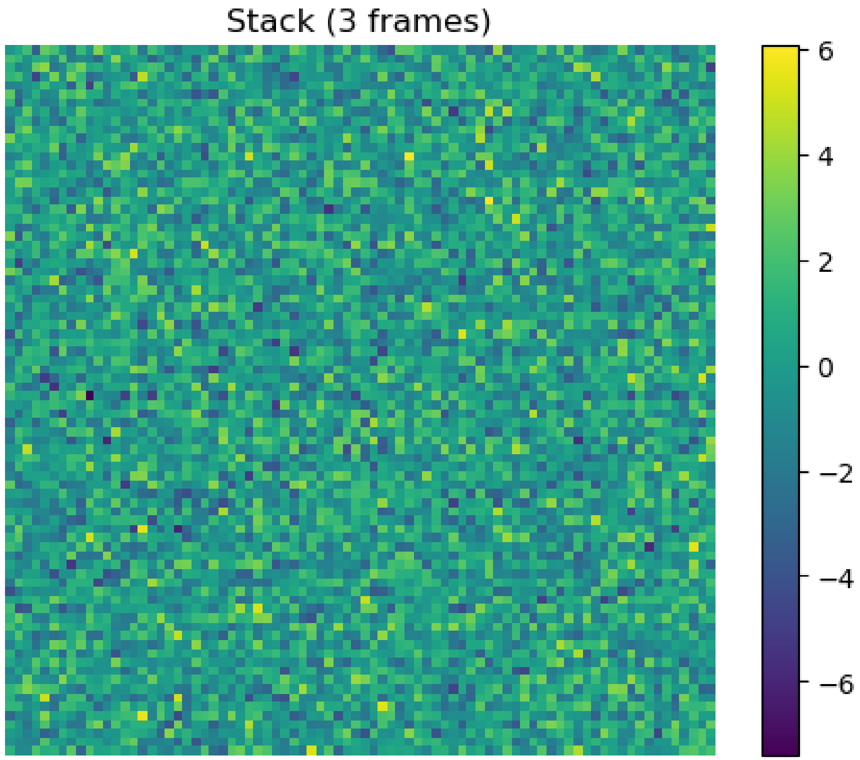


Tested path:
50 pixels South
50 pixels East
(-50,50)

This is an incorrect path

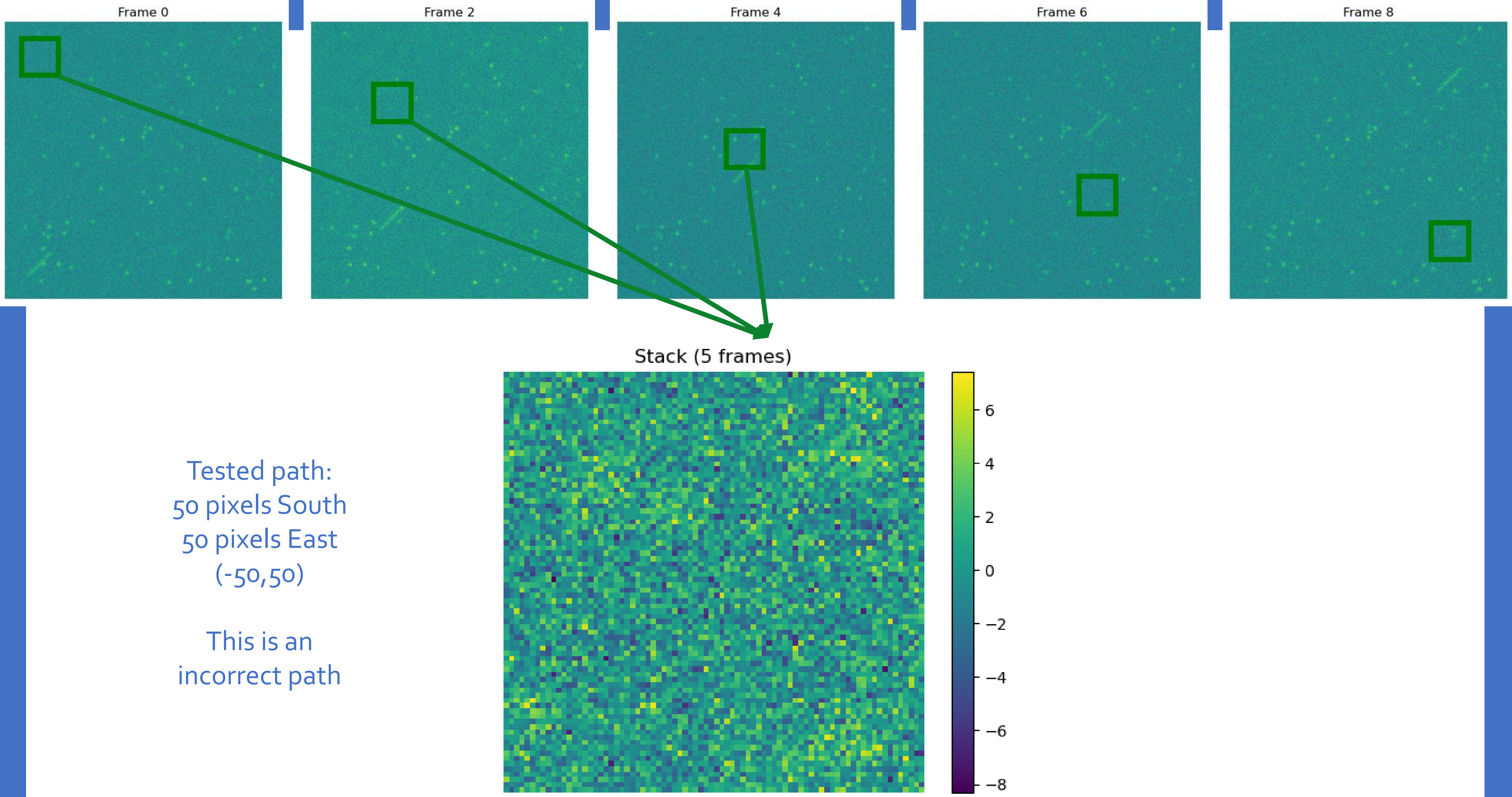






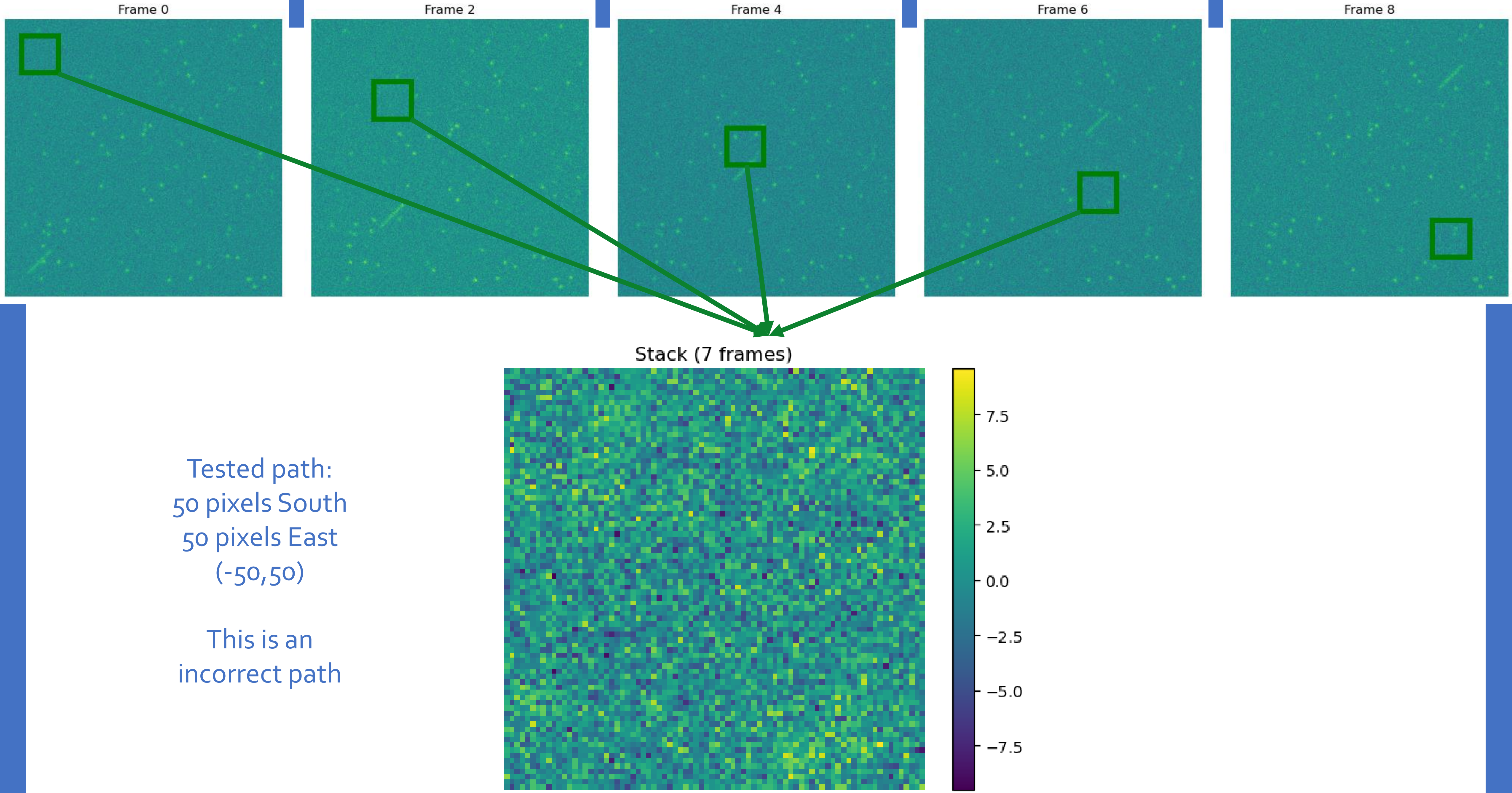
Tested path:
50 pixels South
50 pixels East
(-50,50)

This is an
incorrect path



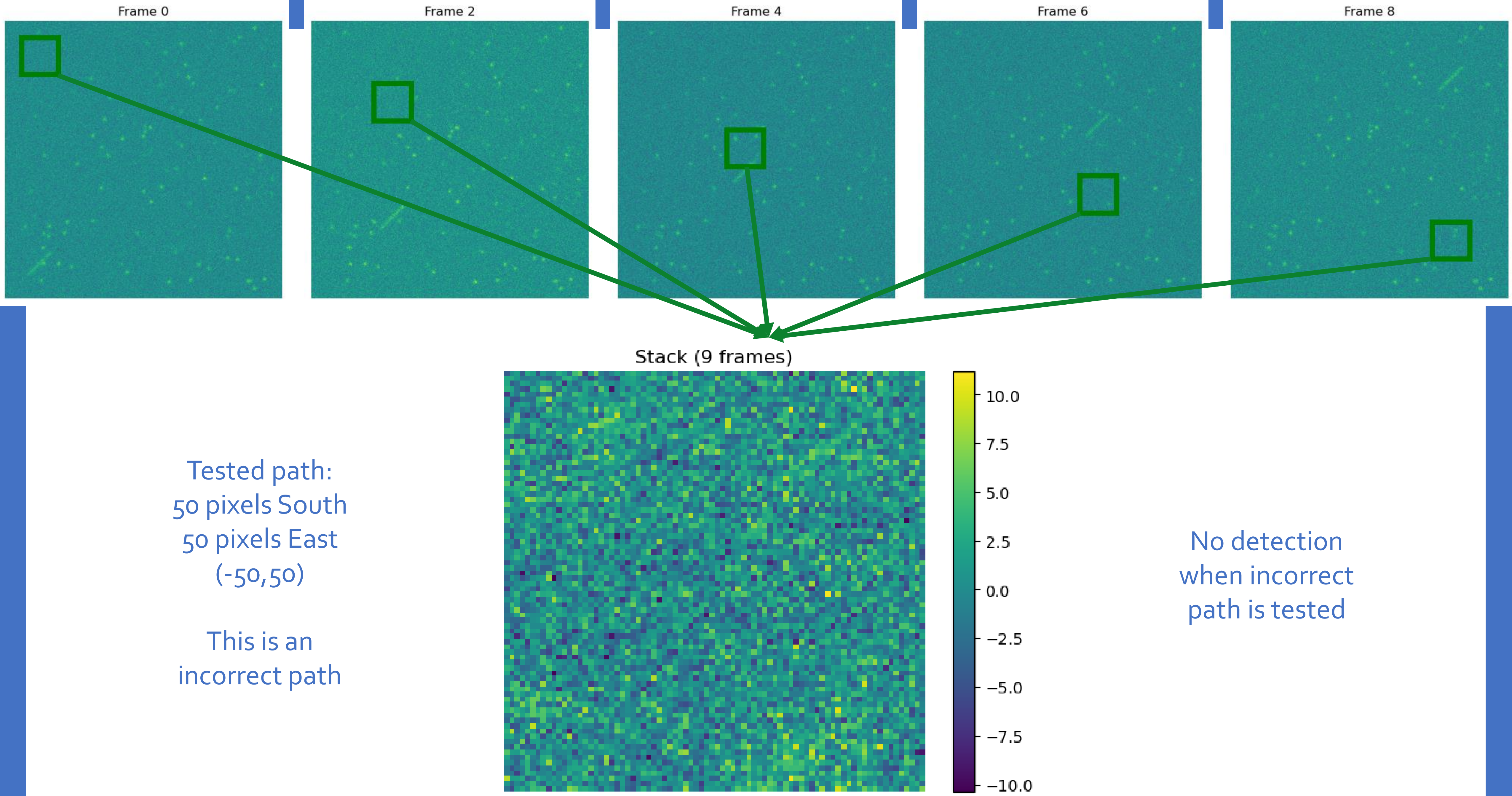
Tested path:
50 pixels South
50 pixels East
(-50,50)

This is an incorrect path



Tested path:
50 pixels South
50 pixels East
(-50,50)

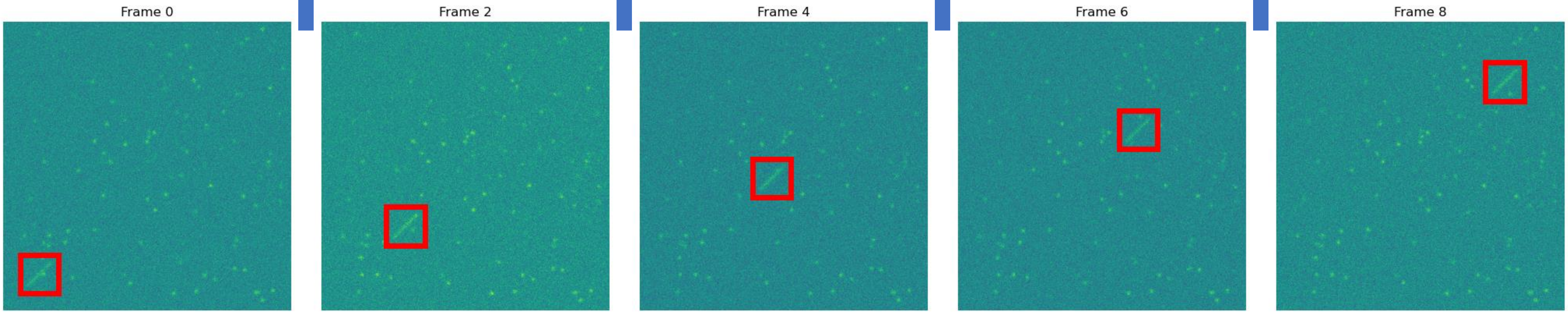
This is an
incorrect path



Tested path:
50 pixels South
50 pixels East
(-50,50)

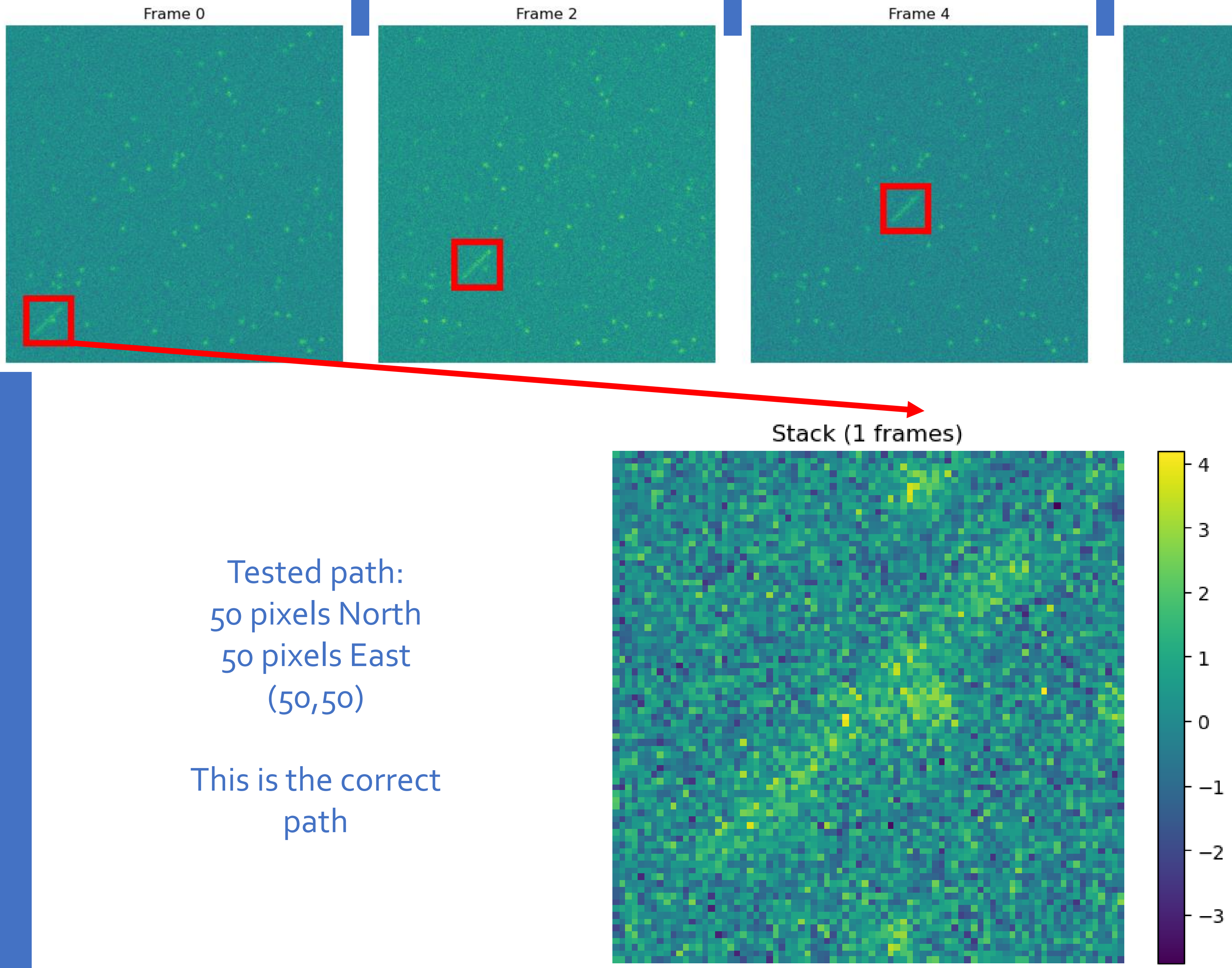
This is an
incorrect path

No detection
when incorrect
path is tested



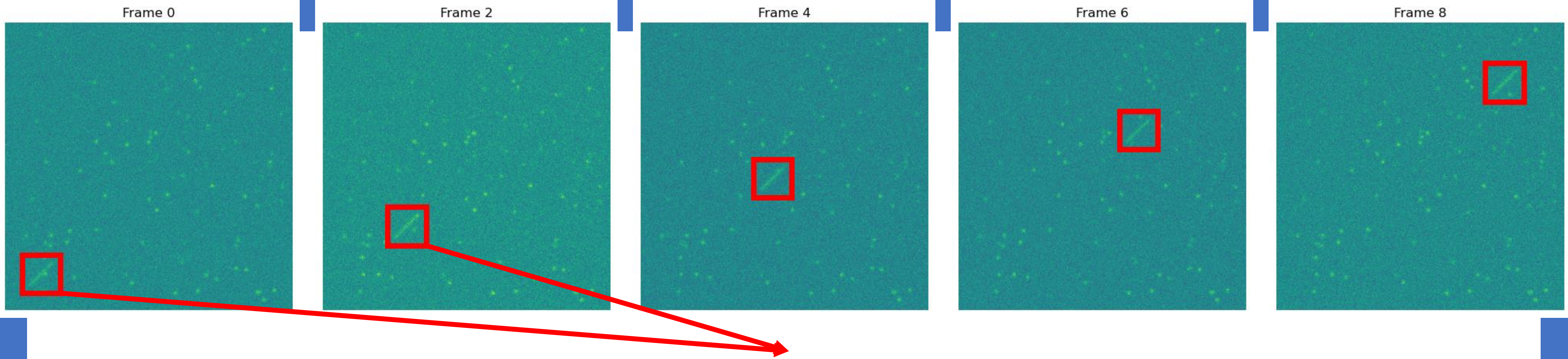
Tested path:
50 pixels North
50 pixels East
(50,50)

This is the correct path



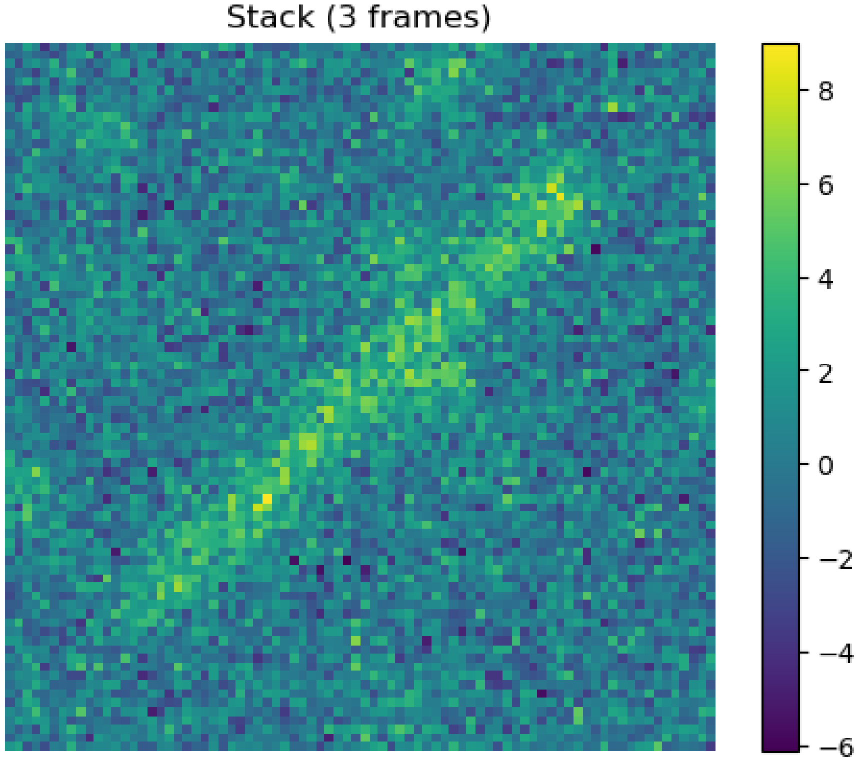
Tested path:
50 pixels North
50 pixels East
(50,50)

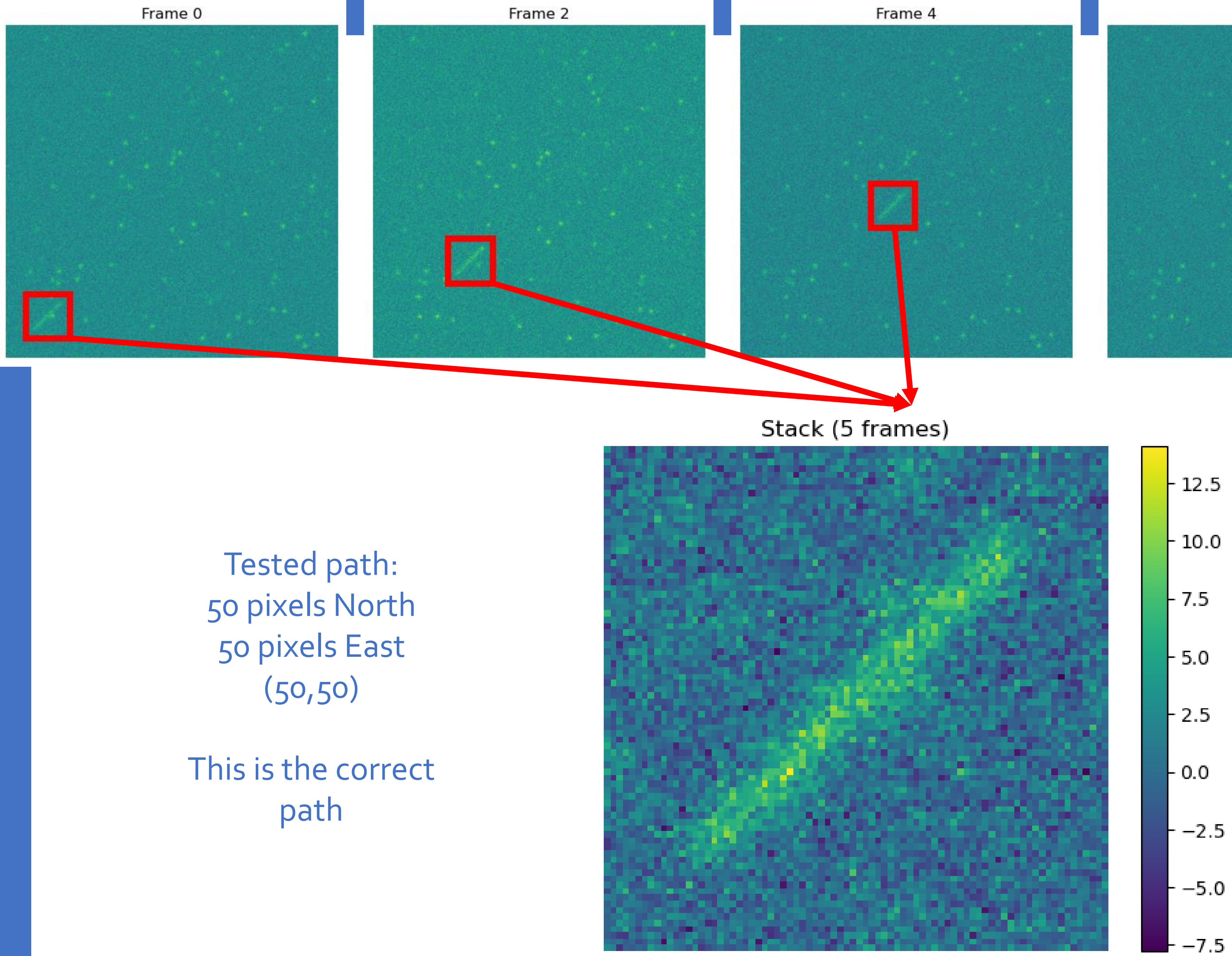
This is the correct path



Tested path:
50 pixels North
50 pixels East
(50,50)

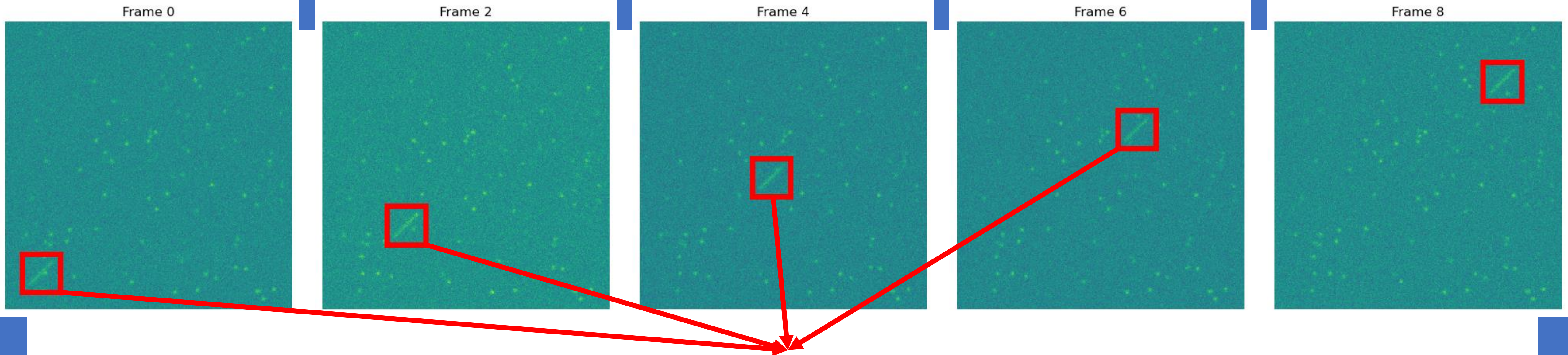
This is the correct path

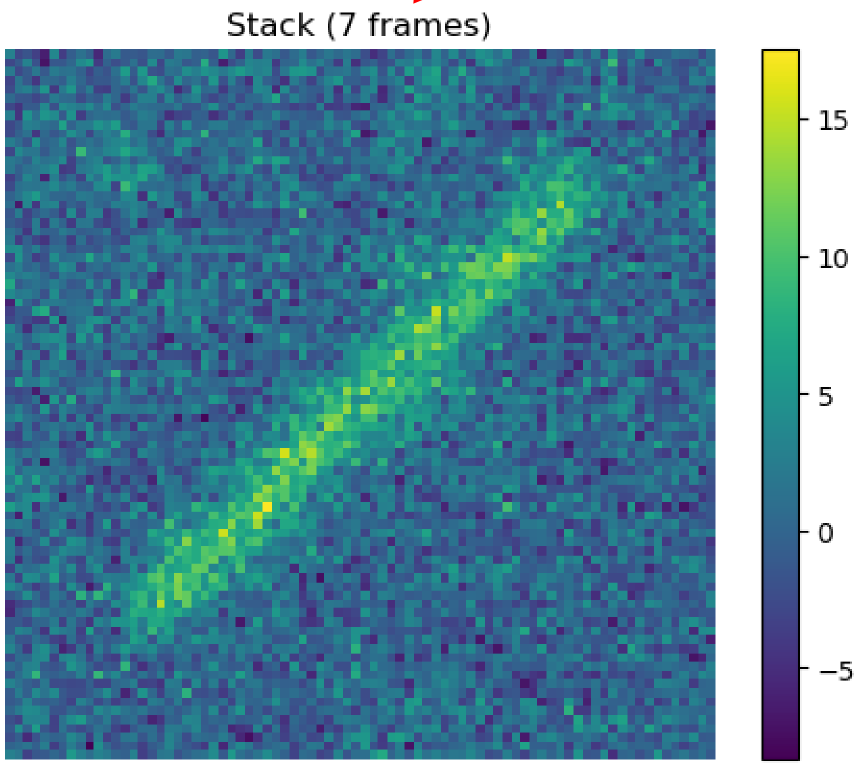




Tested path:
50 pixels North
50 pixels East
(50,50)

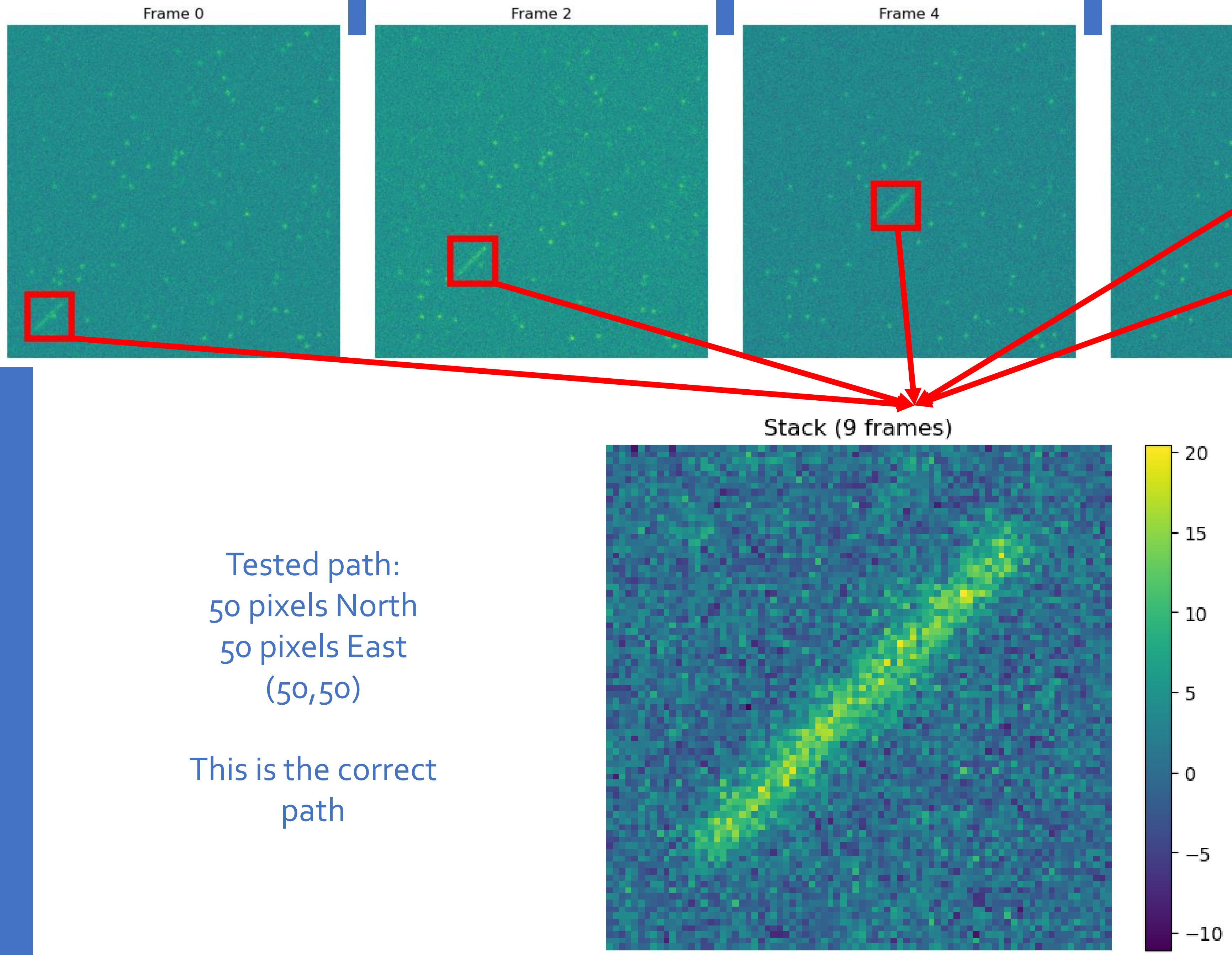
This is the correct path





Tested path:
50 pixels North
50 pixels East
(50,50)

This is the correct path



Tested path:
50 pixels North
50 pixels East
(50,50)

This is the correct path

Very clear detection when correct path is tested

Detection is clearer than in any individual frame

Blind stacking – Pros and Cons

- Pros
  - Can test multiple orbits per dataset
  - Can identify unknown objects or objects on unknown or imprecise orbits
  - Can identify multiple targets per dataset, even if they have different motions/orbits
- Cons
  - Fast objects are seen as extended streaks compared to tracked observations
    - Therefore fainter
  - Computing time is significant

Optimisation avenues

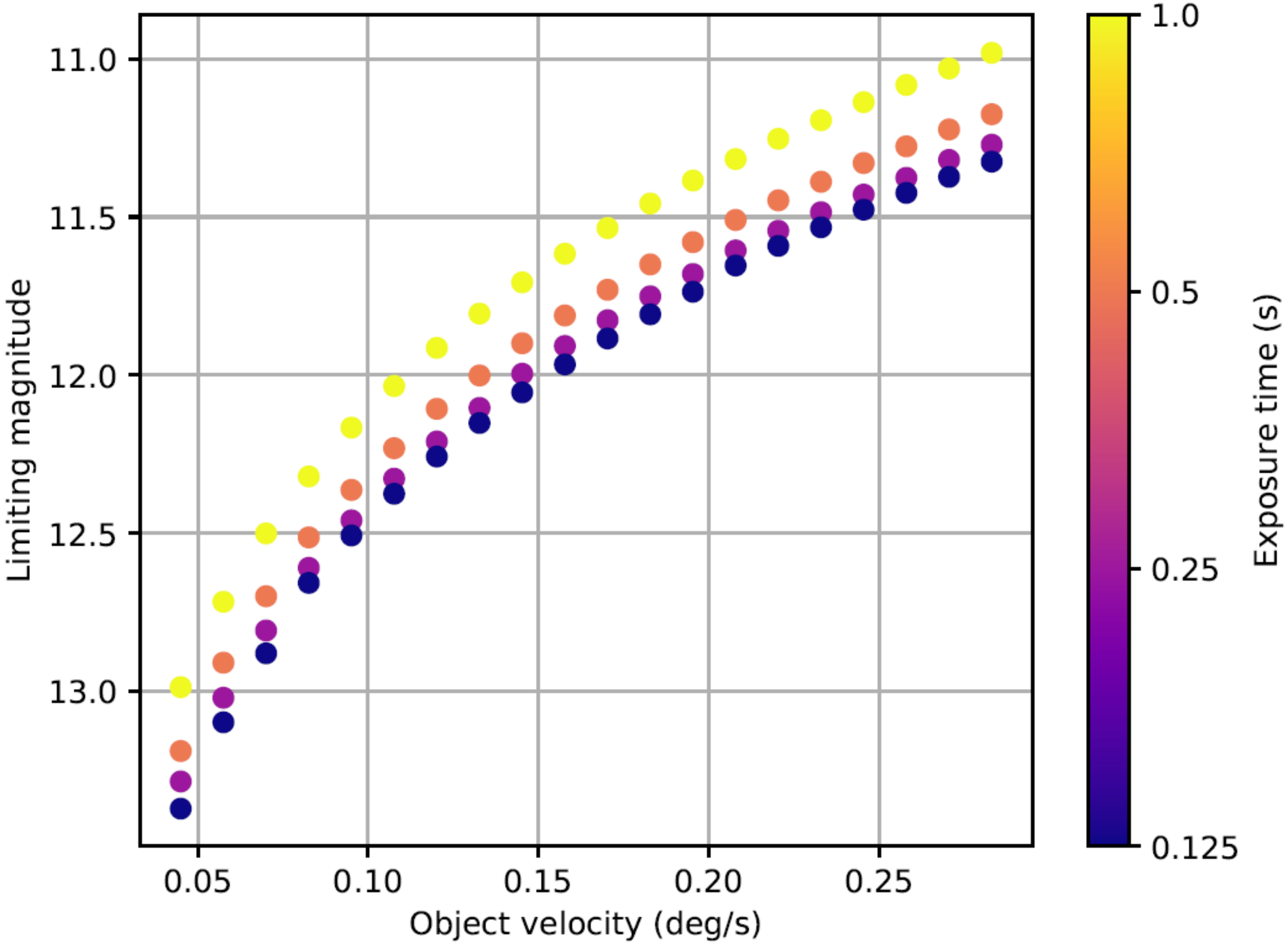
- Exposure time
  - Shorter streaks – correspondingly fewer paths to test
  - Lower background noise (to a point)
  - Greater memory requirement
- Number of frames
  - More signal to stack
  - More memory required
- Binning
  - Increases SNR (to a point)
  - Fewer pixels
  - Less resolution
- Plenty of pipeline optimisations as well
  - Path range, binning style, stacking method etc.

Simulation

- Targets faint enough to test on are hard to reliably schedule
- Need to simulate targets with random speed, direction, magnitude and location
- Based on real data from our telescope in La Palma. Simulated targets are then injected
- Can then run the blind stack code (for a range of optimisation parameters) and check for recoverability

Single frame limits

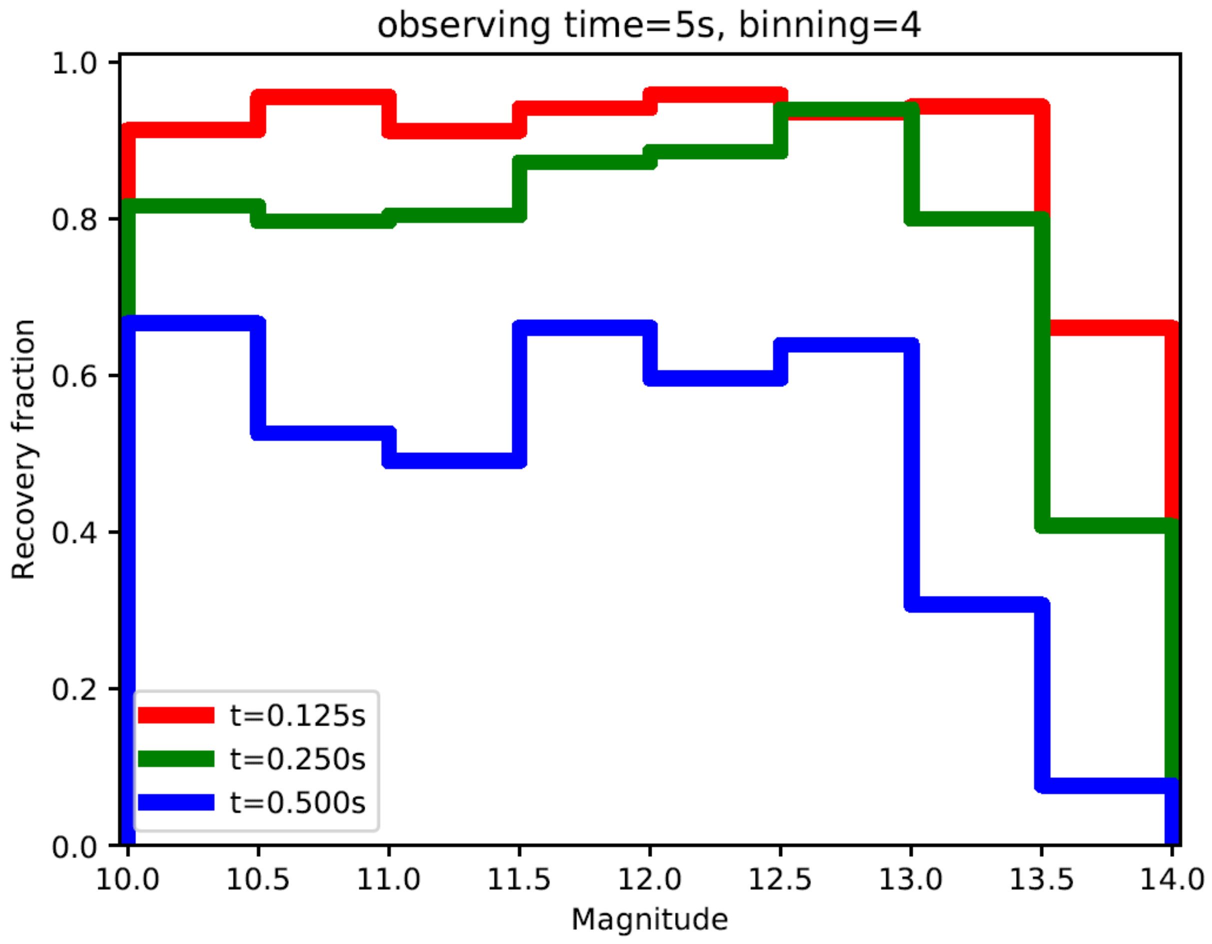
- Based on our observed data we can predict the limiting magnitude of a target in a single frame – function of exposure time and target velocity



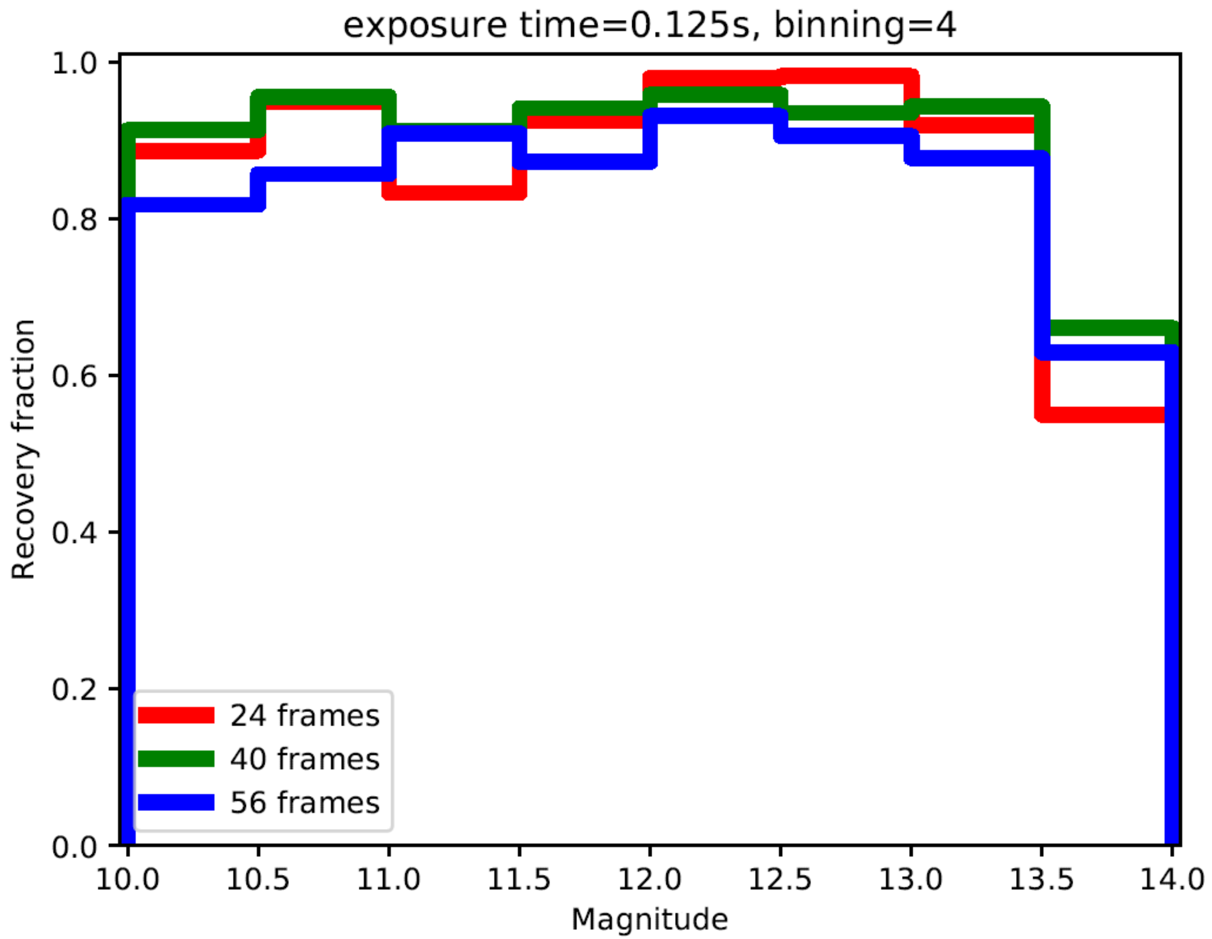
Results – recovery

- Can look at the recoverability of simulated targets as a function of magnitude
- This has been run for multiple combinations of the three optimisation parameters
- In these results we'll look at each optimisation parameter separately

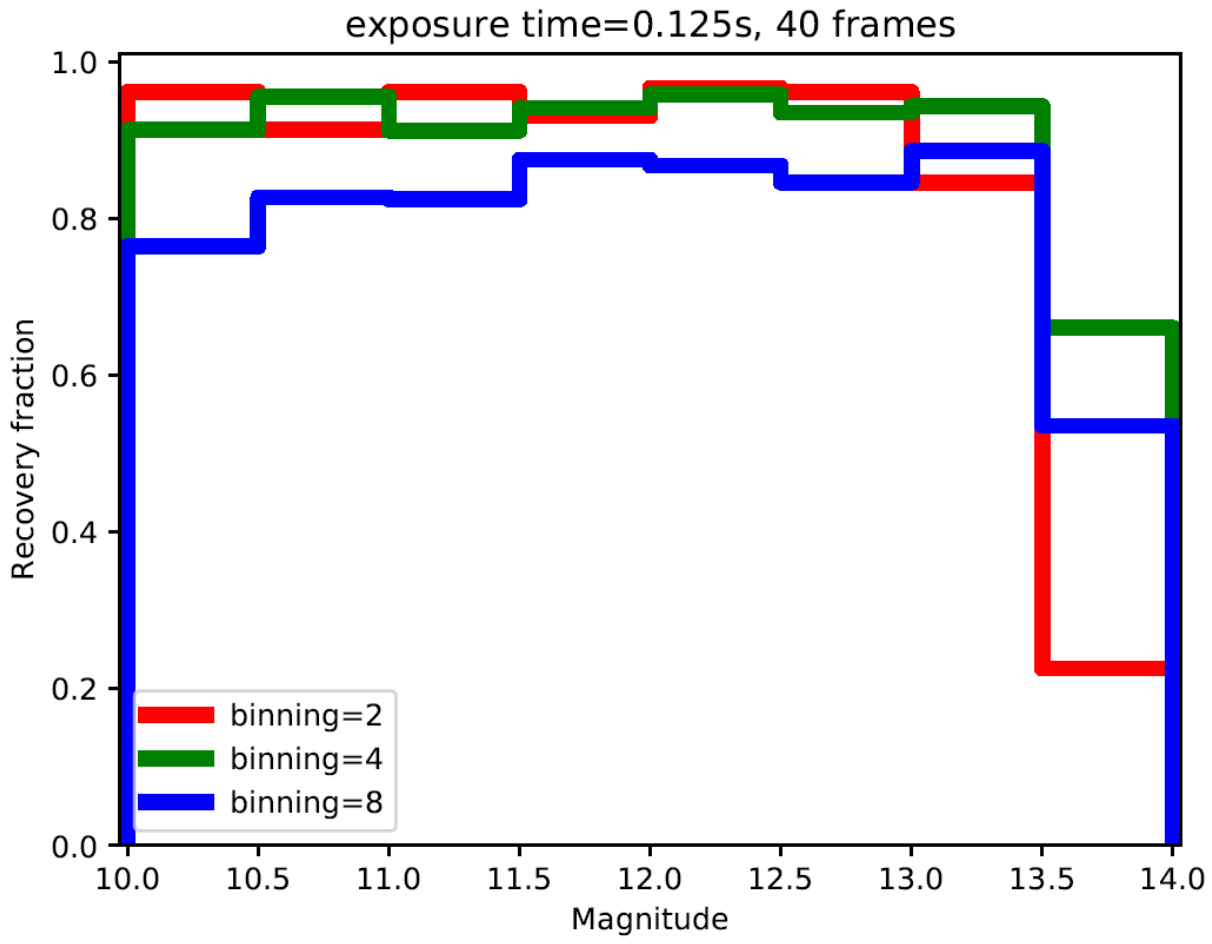
Exposure time



Number of frames



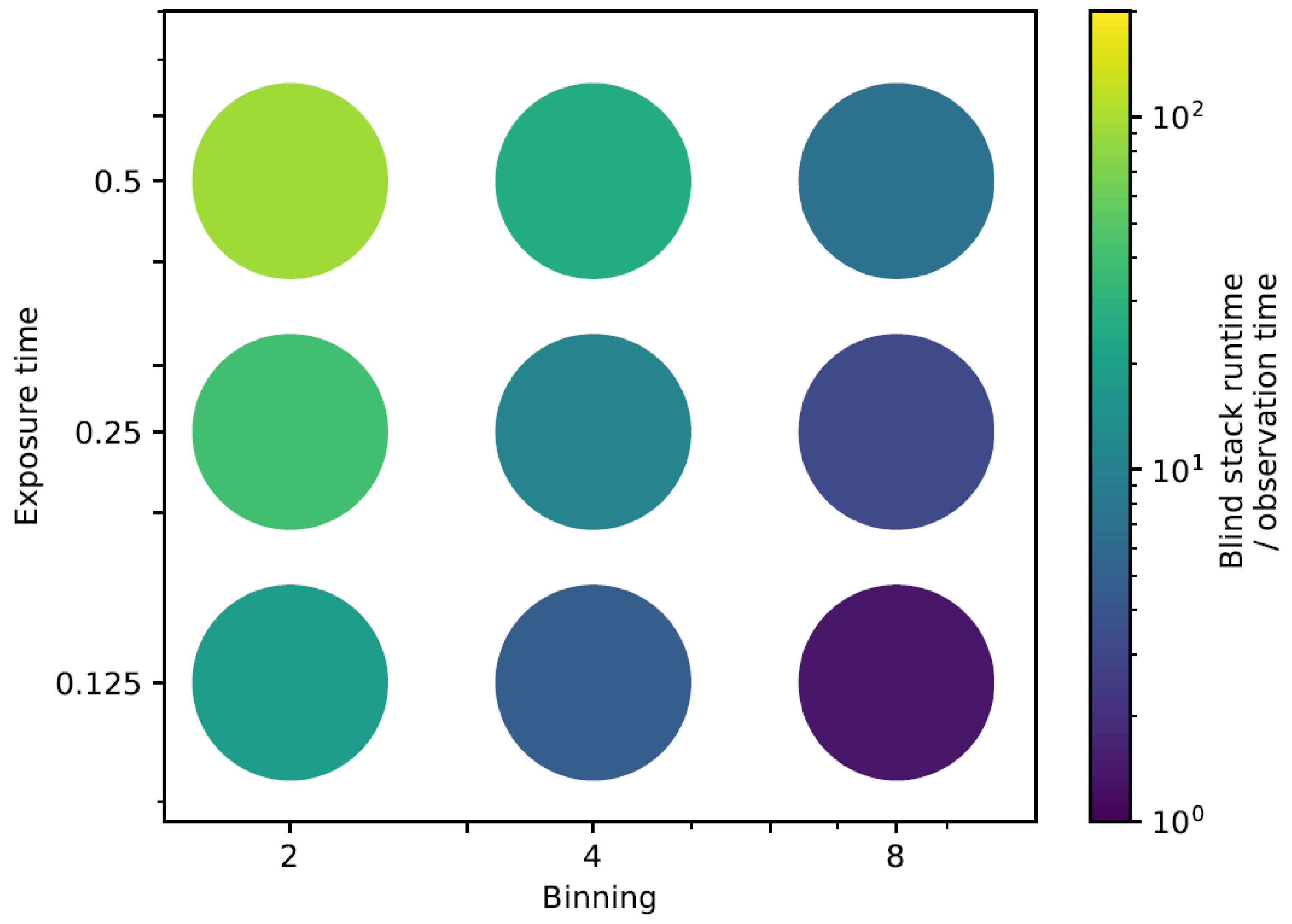
Binning



exposure time=0.125s, 40 frames
Recovery fraction
Magnitude
1.0
0.8
0.6
0.4
0.2
0.0
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0
binning=2
binning=4
binning=8

Results – runtime

- Runtime of the system is also very important
- Ideally want to be able to run in real time
- However, runtime significantly affected by optimisation parameters



Exposure time
0.5
0.25
0.125
Binning
2
4
8
10^0
10^1
10^2
Blind stack runtime
/ observation time

Conclusions and next steps

- Blind stacking can recover LEO objects with little to no prior information
- Can detect fainter objects than are available in individual exposures

- Short exposures are better – lower noise and reduced runtime
- ~5s of exposure total is best – longer and more memory is required, shorter and can't stack as many frames to get the faint targets
- Binning=4 gives best recovery (but binning=8 gives shorter runtime)

- Further computational optimisation
- Continuing comparison with observed data

Thank you for listening