UCLA

UCLA Brain Research Insitute

Cala and noob: Streaming calcium imaging analysis package

*R. W. CHANG[1,2], *J. SAUNDERS[1], T. SASATANI[4], M. BROSCH[1], Z. DONG[1], H. SEMWAL[1,3], P. ZHAO[1], P. GOLSHANI[1], D. AHARONI[1]

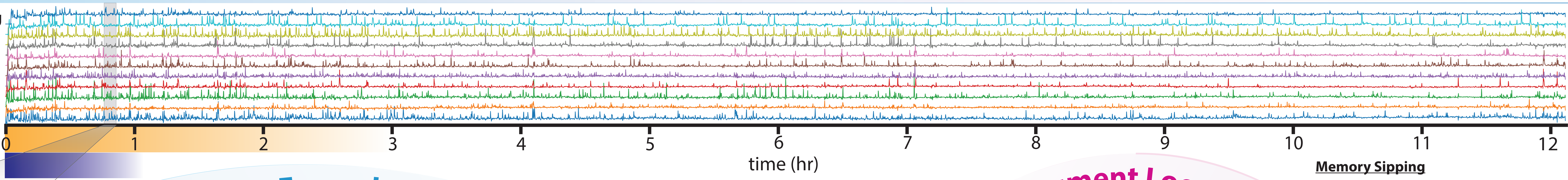
[1]Dept. of Neurol., [2]Physics and Astronomy Dept., [3]Dept. of Bioengineering, UCLA, Los Angeles, CA; [4]Dept. of Electrical Engin. and Information Systems, The Univ. of Tokyo, Bunkyo-ku, Japan

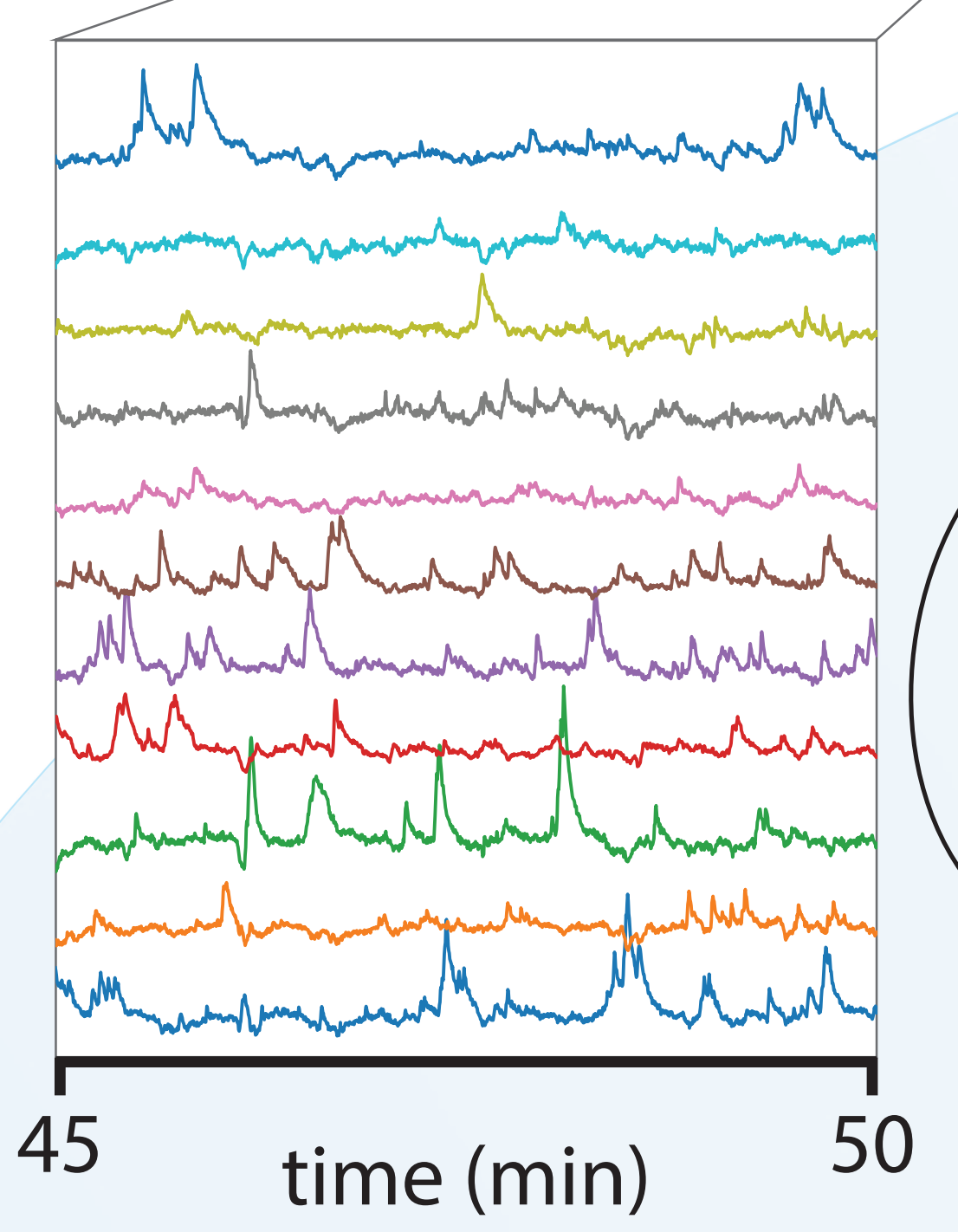
12-hour Recording

Hippocampus CA1
GCaMP6s, Miniscope V4

compute-intensive software[1]

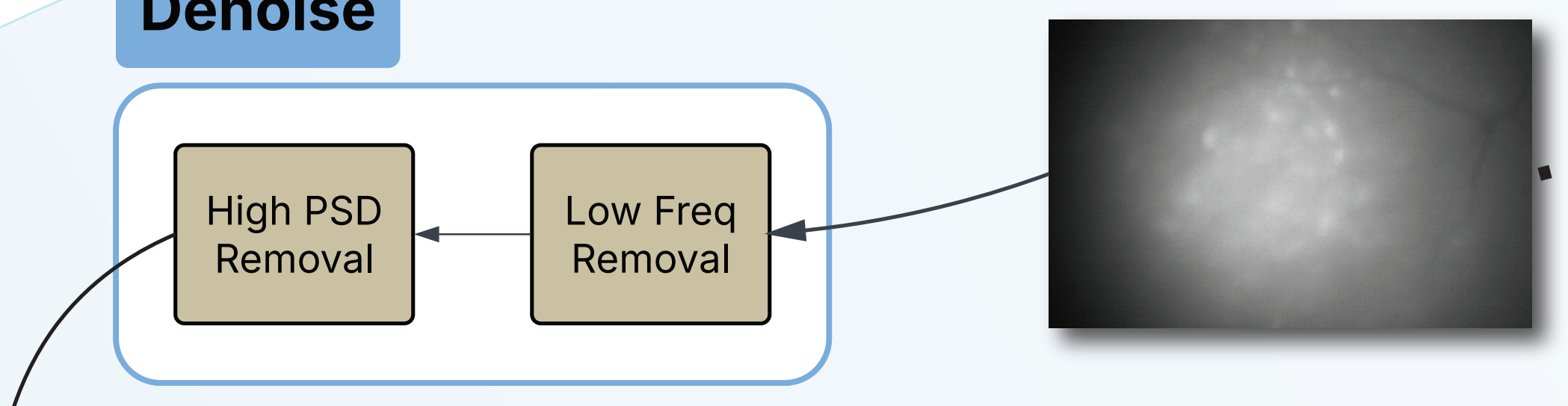
memory-intensive software[2]



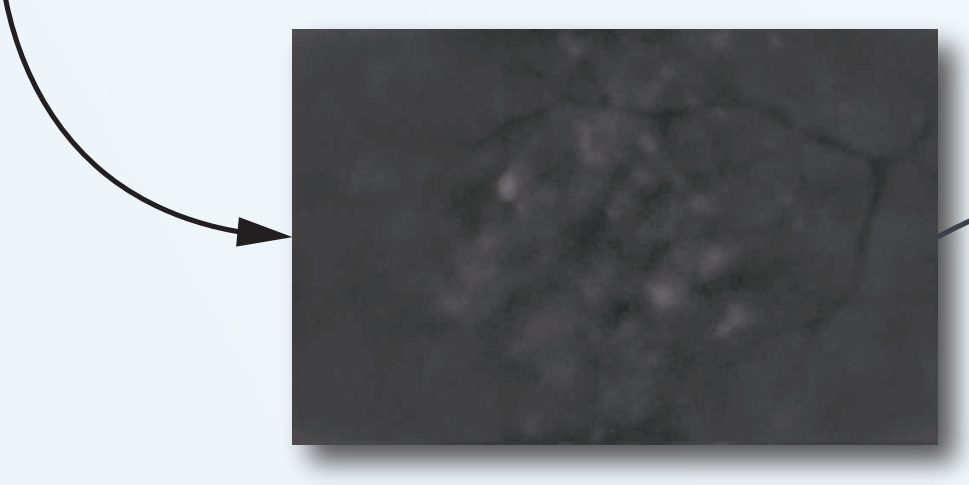


Frame Loop

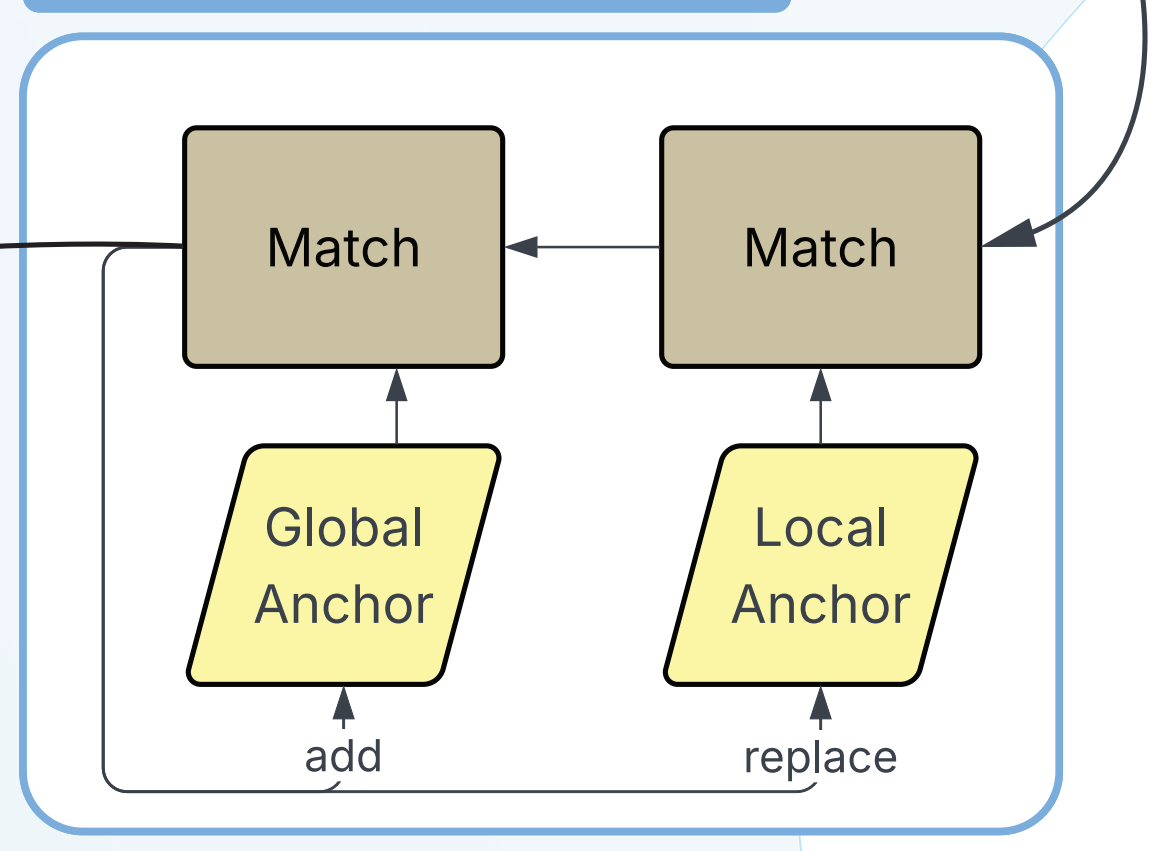
Denoise



Done entirely frame by frame with frequency filtering.



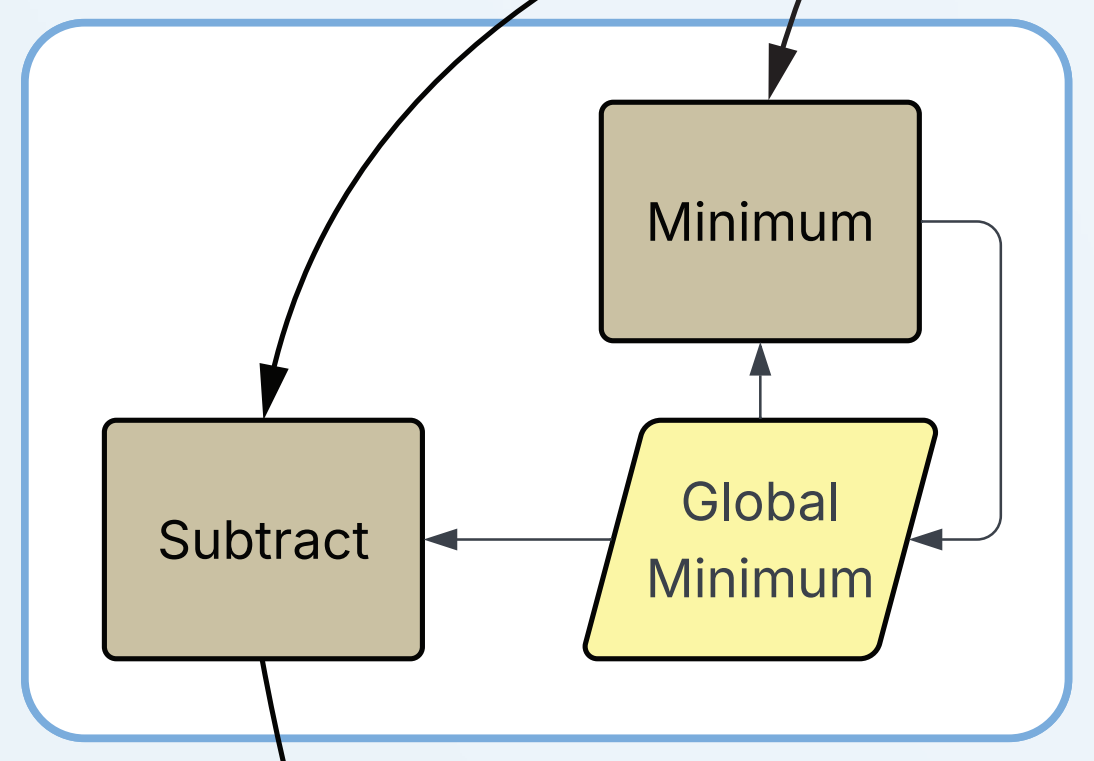
Motion Correction

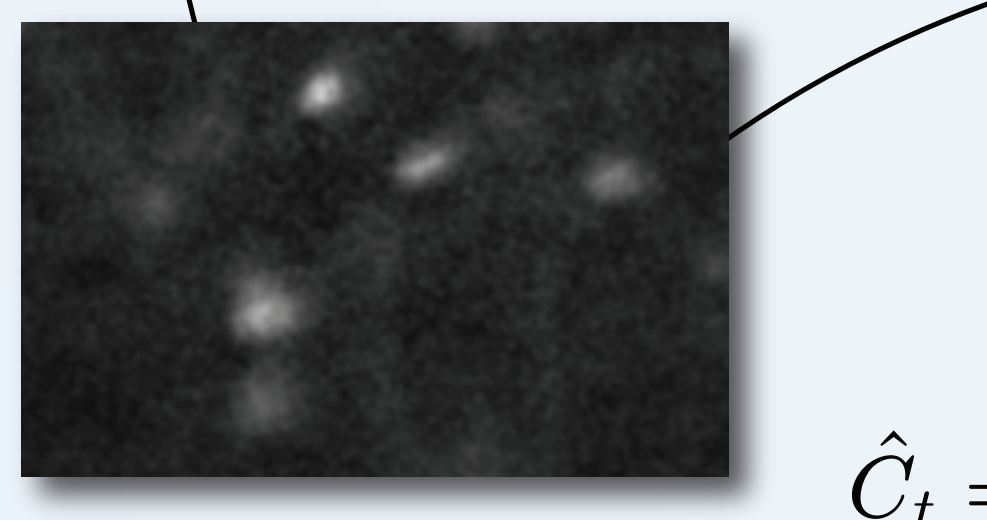


Motion correction has two stages to account for limitations from streaming. Global anchor acts as a true center, but may lack temporally local features. Local anchor becomes the bridge between the two.

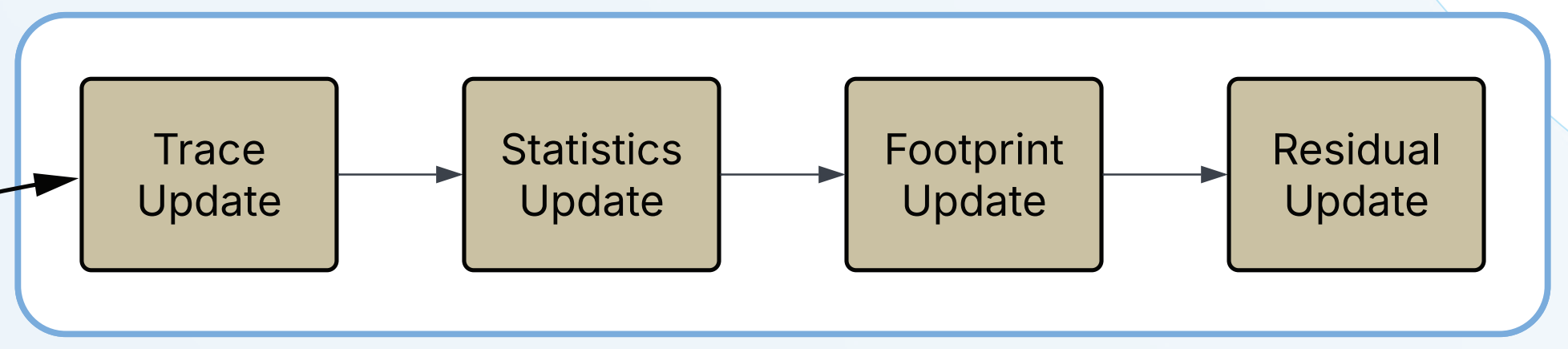
Deglow step removes background, which becomes one less thing for the main algorithm to worry about.

Deglow

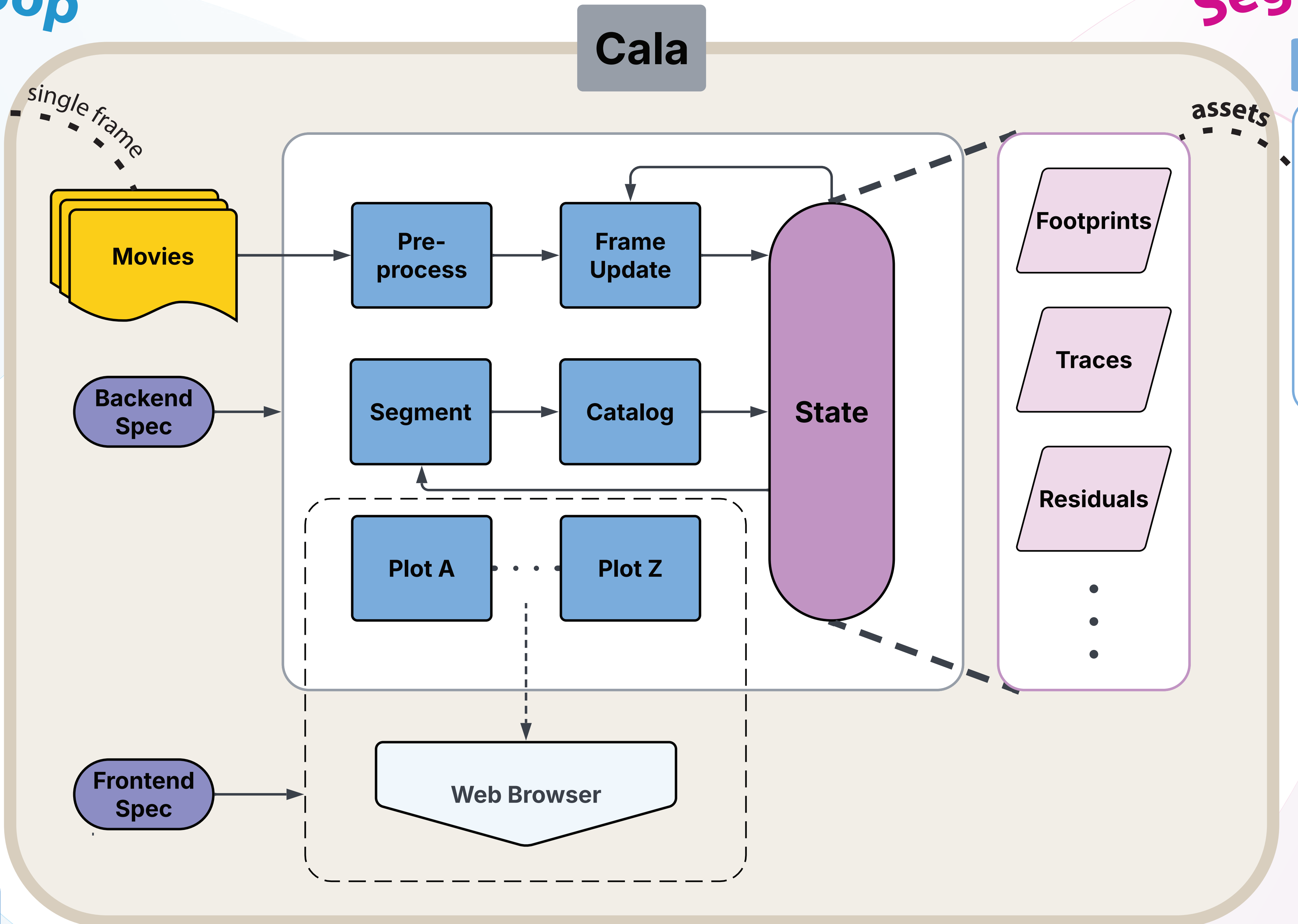




Frame Update

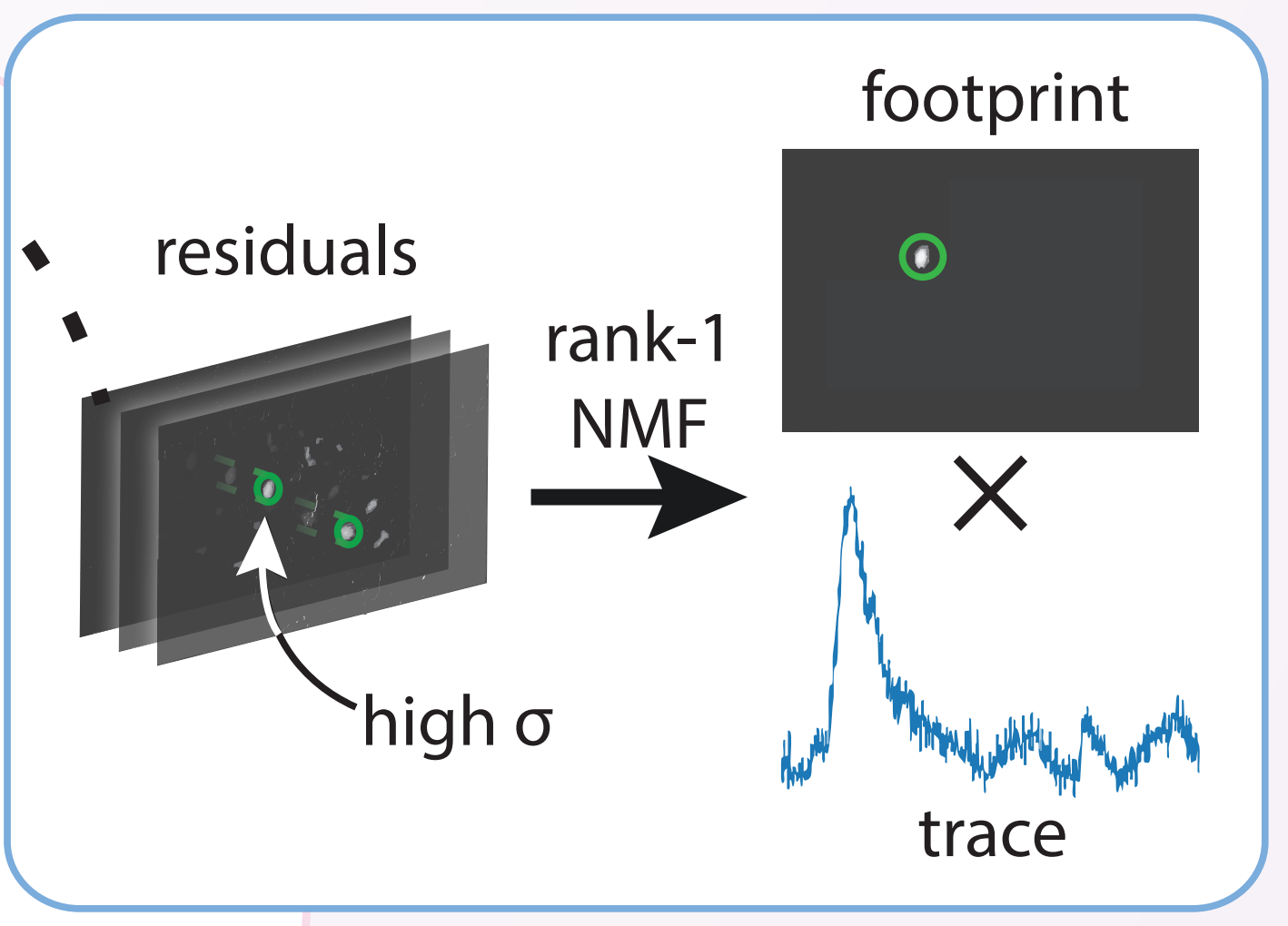


Cala



Segment Loop

Segment



Decomposition[3]

1. From a stack of **residuals**, slice out the area with standard deviation above threshold.
2. Peform rank-1 NMF.
3. If Frobenius norm of the matrix difference is below threshold, accept.

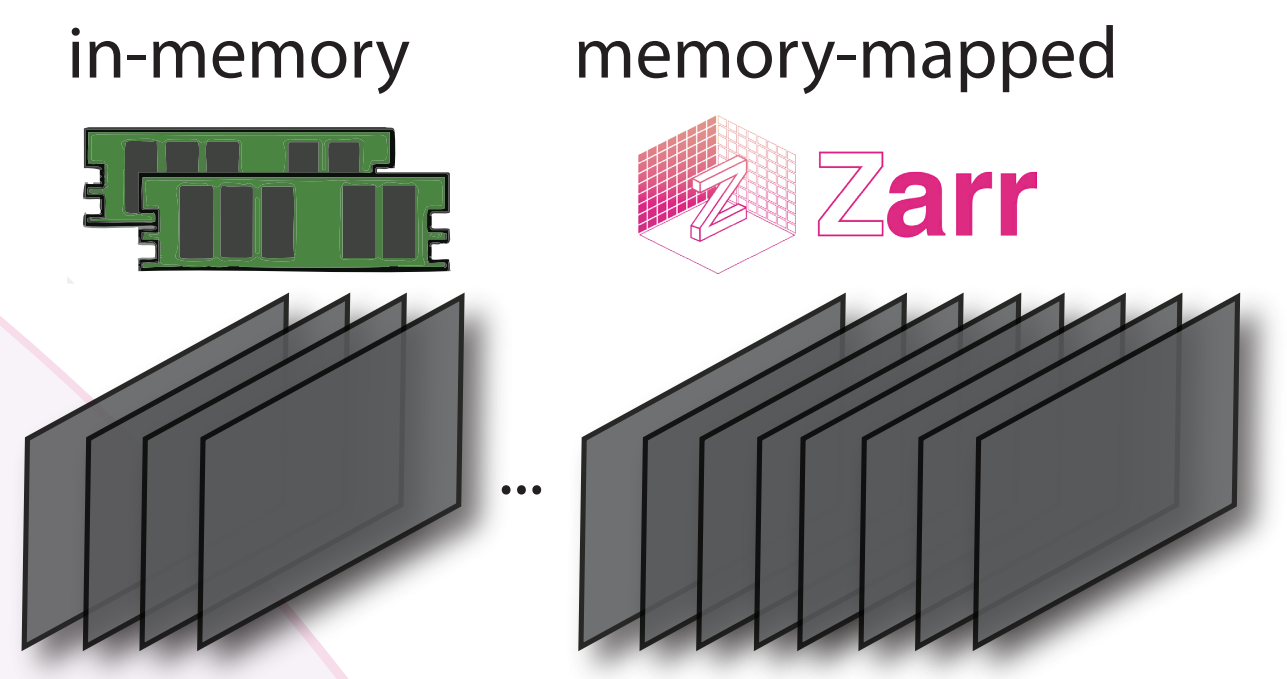
Merge

If two **estimators** have...
1. overlapping footprints
2. traces correlating above threshold

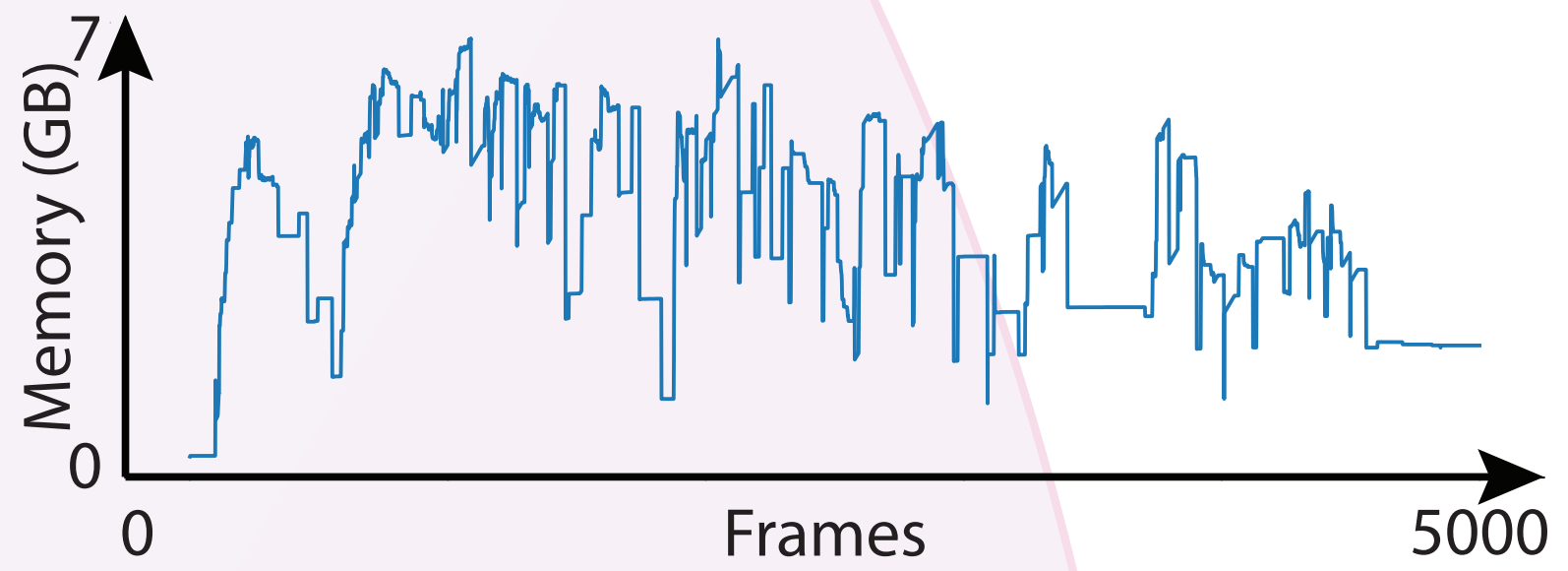
then they are merged.

Afterwards, we filter the estimators by their quality (e.g. footprint area >> mean footprint area).

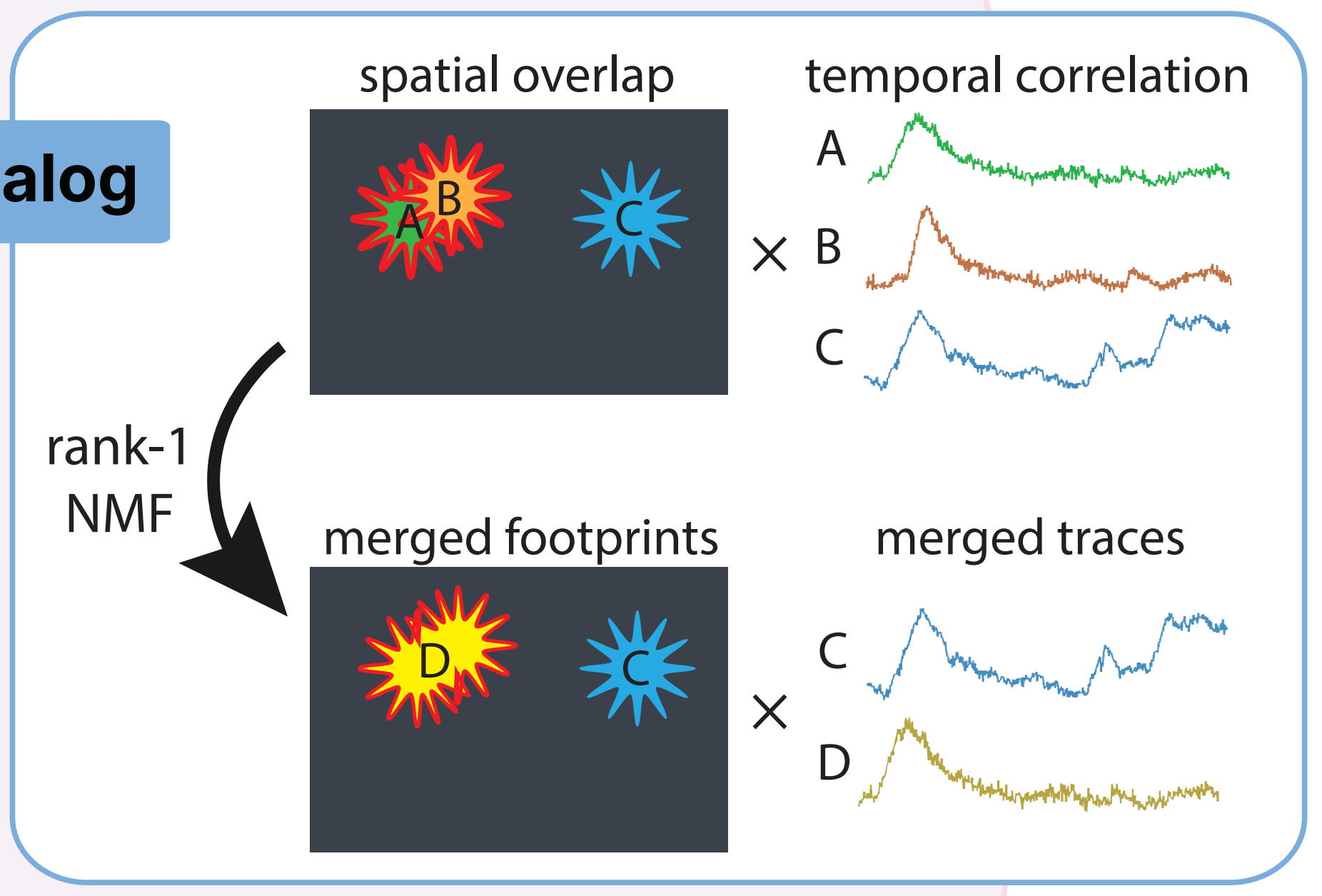
Memory Sipping



For growing **assets**, we only keep necessary amounts in-memory, while flushing the rest to storage. This achieves the complete stable memory demand throughout analysis observable in the graph below.



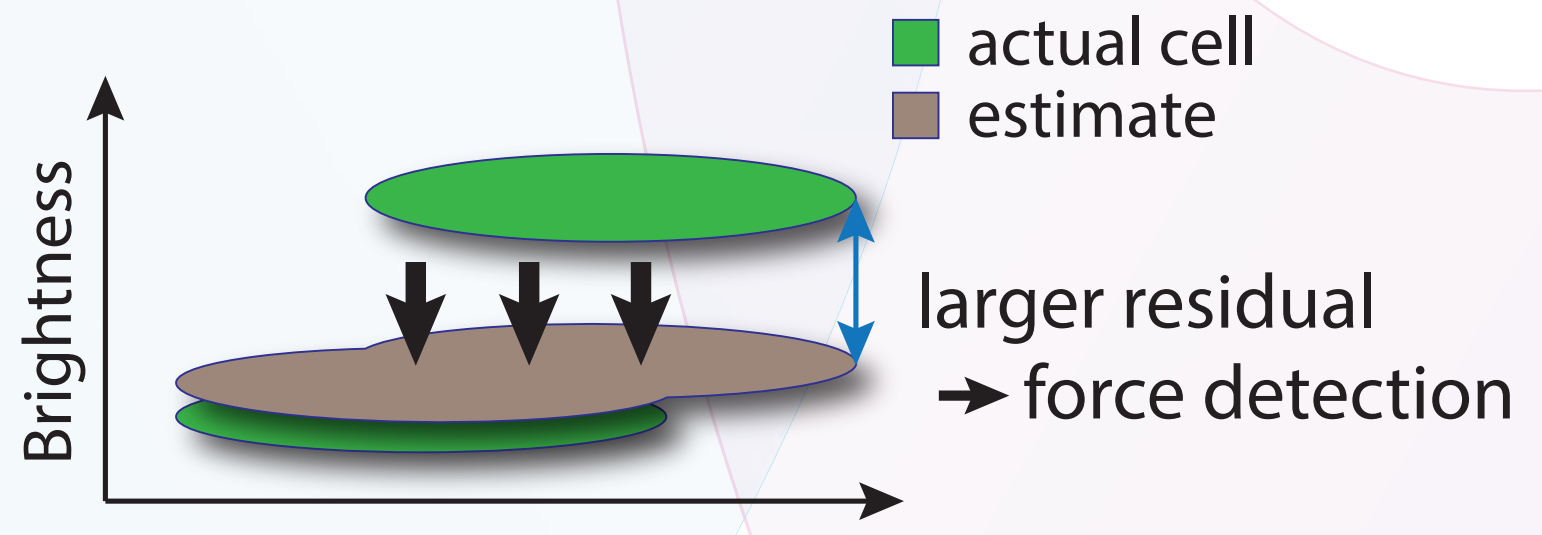
Catalog



Residual: The Unexplained

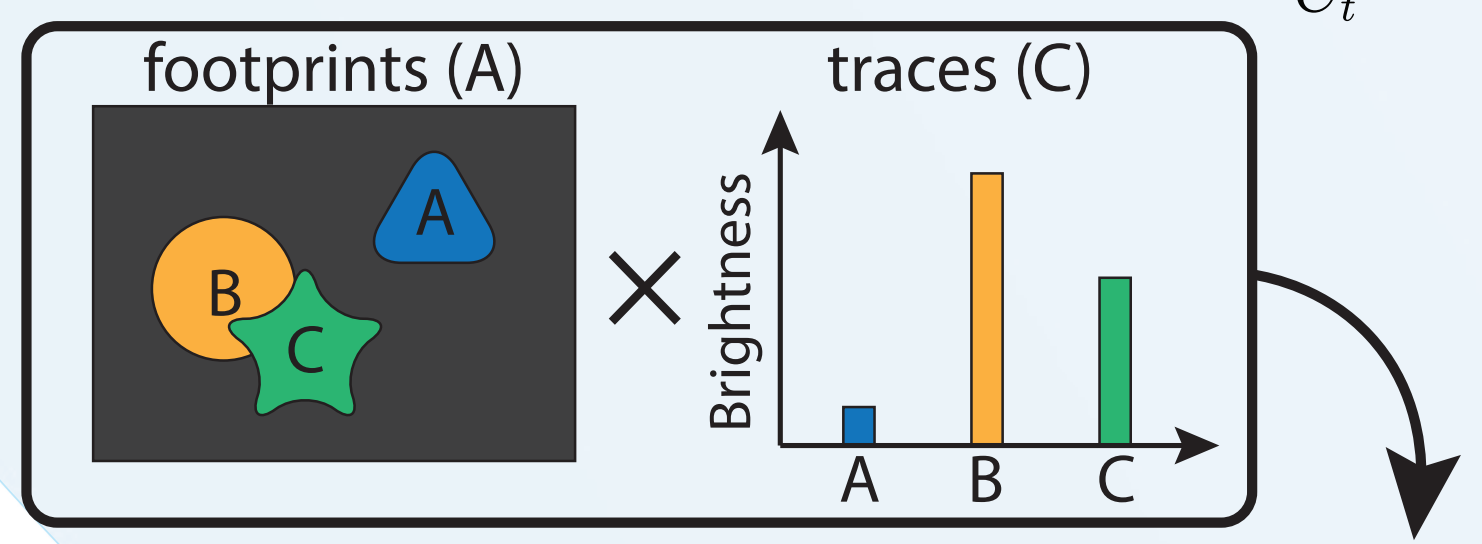
$$R = Y - AC$$

Lastly, we have the "unexplained" after everything currently known has been updated. This will be later used in segment loop to find new estimators.

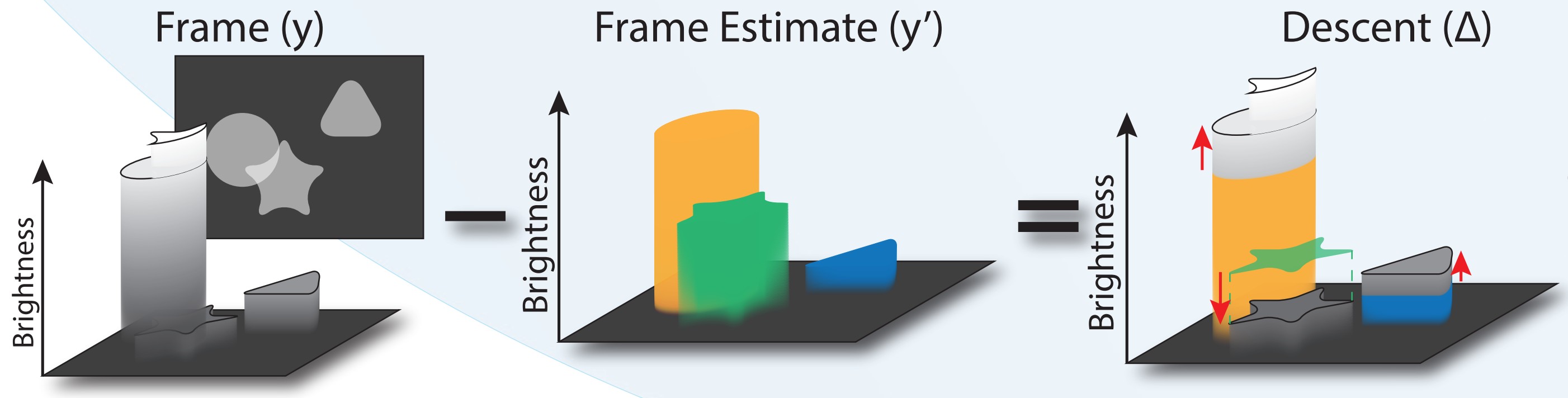


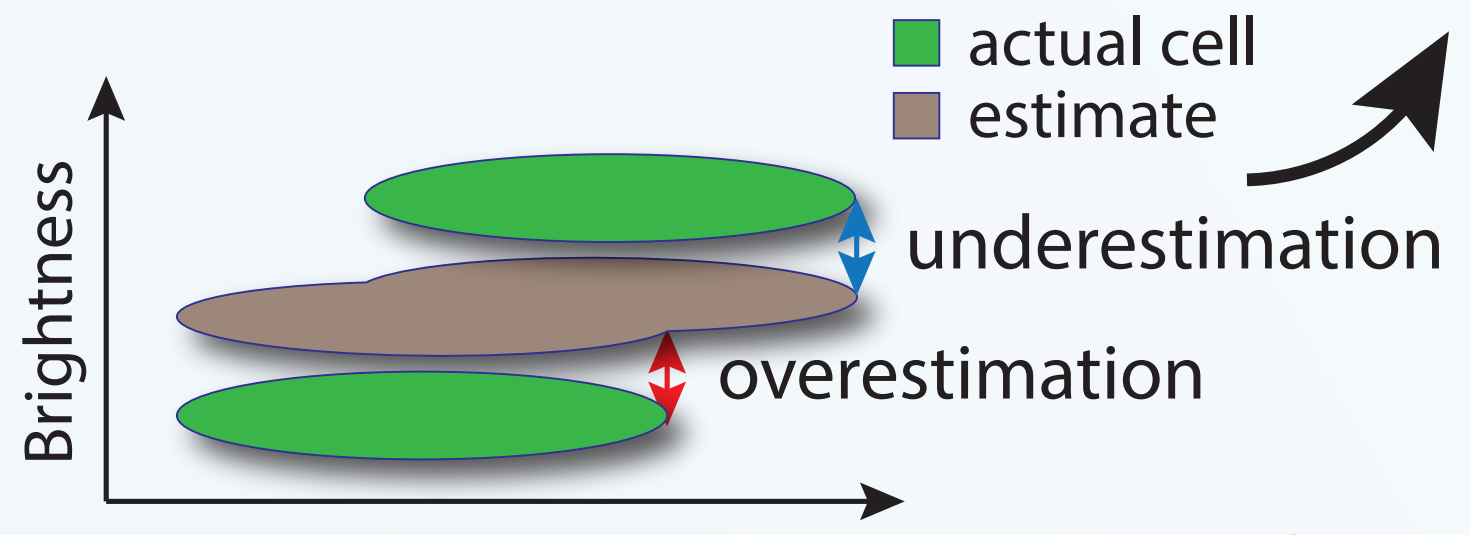
Online Matrix Factorization[3]

$$\hat{C}_t = \underset{C_t}{\operatorname{argmin}} \, ||Y_t - AC_t||^2 \qquad \hat{A} = \underset{A}{\operatorname{argmin}} \, ||W - AM||^2$$



These two equations are the meat of the OMF algorithm. In essence, they work identically - they find the difference between their estimate of the "frame" and the actual "frame," and add the difference back to their estimate.





Separation Anxiety

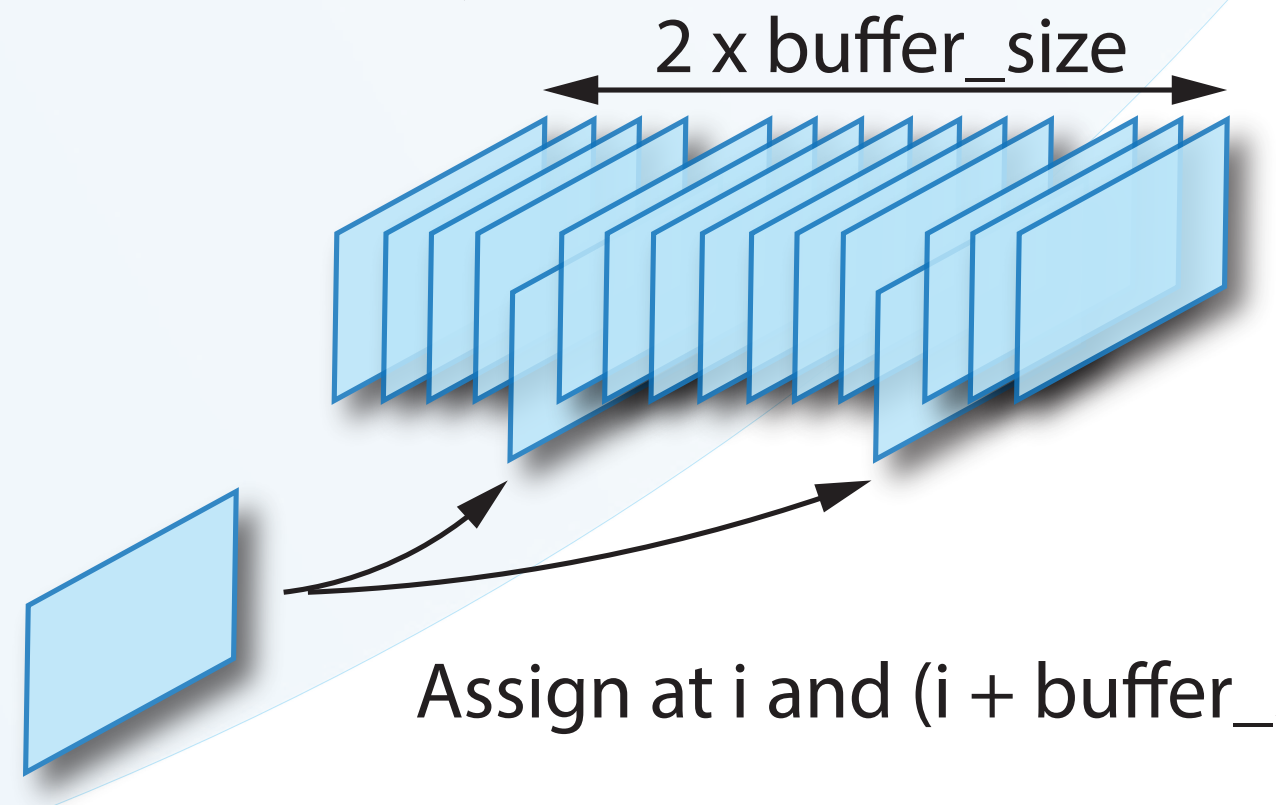
Underfitting (**estimator** needs to be split) is addressed by forcing the detection of another **estimator**, by penalizing overestimation more than underestimation.

Statistics: Long Exposure

$$W = YC^\top \qquad M = CC^\top$$

"frame summary" "trace summary"

These are the **Statistics**. They compensate for the lack of all the frames in memory. You could think of them as "component-pixel correlation (W)" and "component-component correlation (M)."



Buffers with a View

Bip-buffer means **assets** bypass array deep-copy unlike circular buffer, since it always returns contiguously sorted result.

Assign at i and (i + buffer_size) mod (2 x buffer_size)

~ noob ~

noob is a graph processing library for scientists. Pipelines that scale down to a single function call and scale up to your massive compute cluster. Bring your existing code, make no changes, build simple workflows and glue them together into something greater than the sum of its parts.

nodes

Pipelines are graphs of nodes. Each node can have **slots** (inputs) and **signals** (outputs) that correspond to the arguments and return values of their underlying callable.

e.g. here a miniscope might return a frame and metadata from each call, and accept some optional "stim" parameter to control optogenetic stimulation

nesting

Pipelines can take inputs and return values, so they can be nested as nodes within other pipelines.

Nesting pipelines gives a natural means of controlling data locality, process boundaries, and composing multiple pipelines!

cardinality

Things are never just single things: noob has built-in support for cardinality manipulation for concurrency and to glue unwilling nodes together

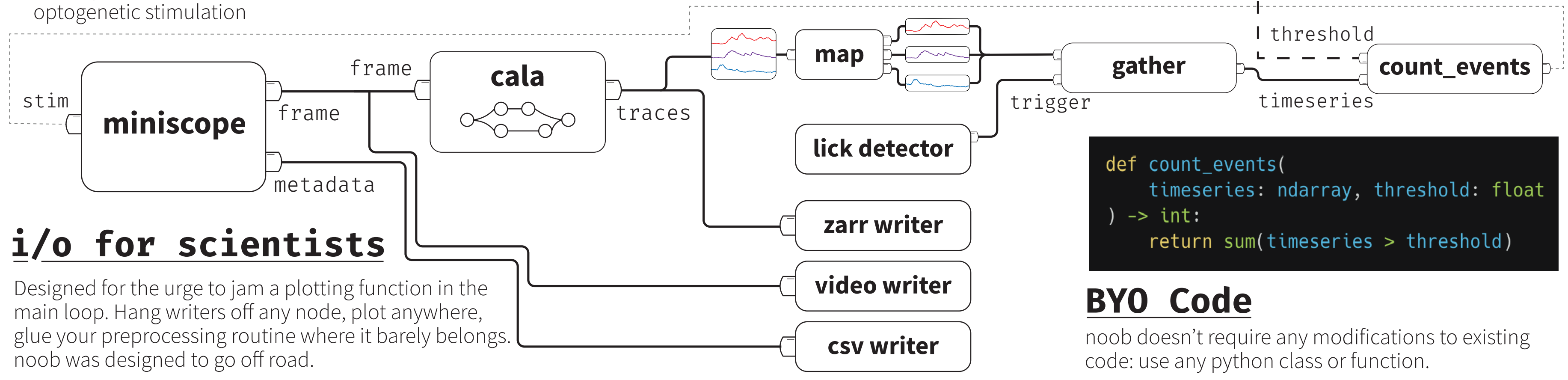
map: split a single event into multiple events to process concurrently

gather: combine multiple events into one, by batch or with an external trigger.

declarative

Mix dynamic, event-based data with static data - the pipeline is self-contained and publishable.

```
counter:
  type: mypackage.count_events
  params:
    - threshold: 0.5
  depends:
    - timeseries: gather.value
```



```
def count_events(
    timeseries: ndarray, threshold: float
) -> int:
    return sum(timeseries > threshold)
```

i/o for scientists

Designed for the urge to jam a plotting function in the main loop. Hang writers off any node, plot anywhere, glue your preprocessing routine where it barely belongs. noob was designed to go off road.

BYO Code

noob doesn't require any modifications to existing code: use any python class or function.

[1] A. Giovannucci, et al. (2019) eLife 10.7554/eLife.38173;
[2] Z. Dong, et al. (2022) eLife 10.7554/eLife.70661;
[3] A. Giovannucci, et al. (2017) NeurIPS 10.5555/3294996.3294999;
[4] J. Friedrich, et al. (2020) PLoS Comput Biol 10.1371/journal.pcbi.1008565;

This work was supported by the following grants:
NIH DP2 MH129986, and NIH U01 1NS122124.
We thank the BRAIN Initiative and NIH for the
continued support of the UCLA Miniscope Project.