

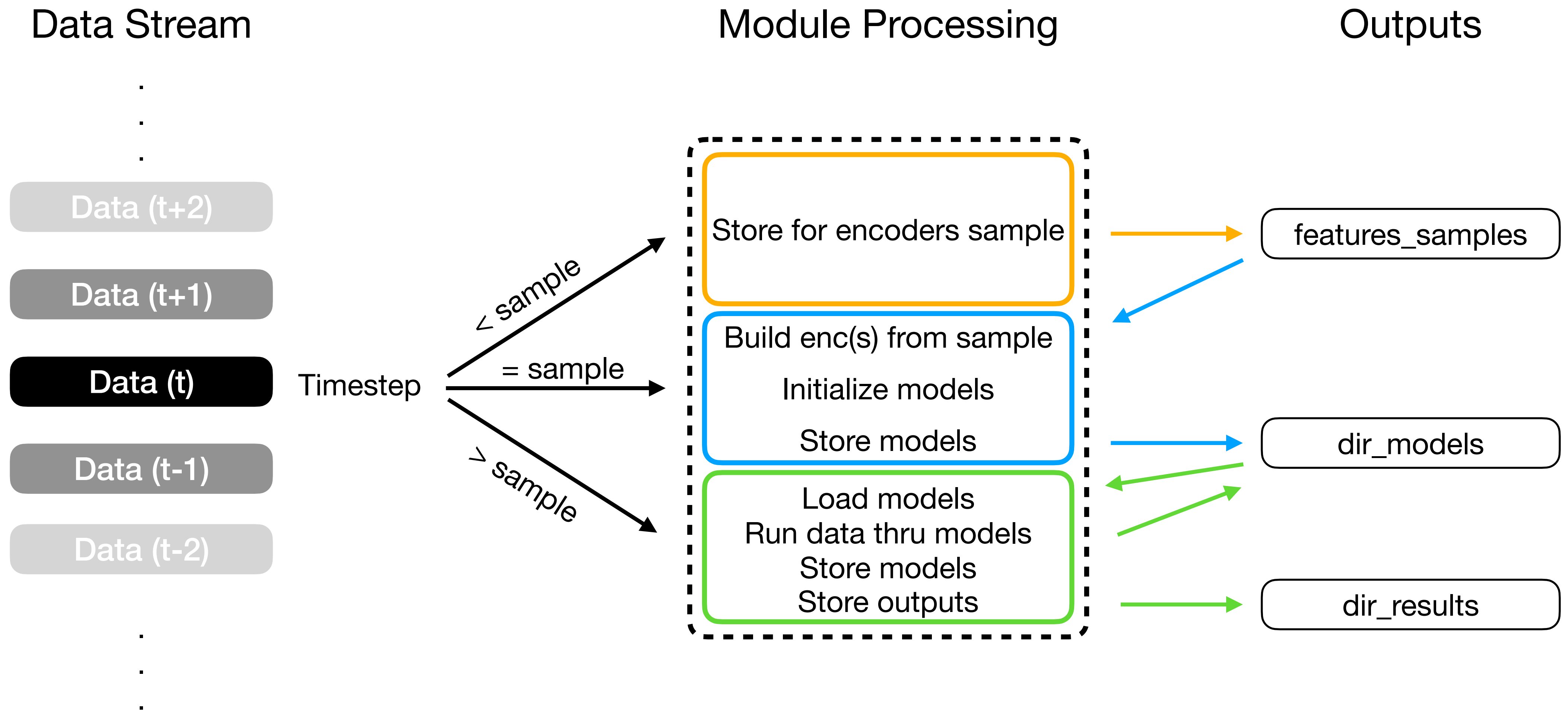
# HTM.core Streamer

Python Module

Sam Heiserman — Jan, 2022

# Module Overview

1) Sample 2) Initialize 3) Run



# Pseudocode

## Main Functions

### htm stream runner

1. Load --> Config
2. Load —> Batch Data
3. For Row in Batch Data:
  1. Generate —> Stream Data
  2. Store —> Stream Data
  3. Run —> **stream\_to\_htm()**

### stream to htm

1. Load —> Config
  2. Load --> Stream Data
  3. Validate Config
- if mode == **Sample**:
4. Store Data
- elif mode == **Initialize**:
4. Store Data
  5. Build Encoder Params
  6. Initialize HTM model(s)
  7. Store HTM model(s)
- else: (mode=**Run**)
4. Load HTM model(s)
  5. Run Stream Data thru HTM model(s)
  6. Store HTM models() outputs
  7. Store HTM models()
  8. Store Config

# Config Structure

## Set by user

Where on disk is data & models stored?

```
dirs:
  data: /where/to/stream/to
  models: /where/to/save-load/models
  results: /where/to/save/results
```

Which features are modeled?

```
features:
- Solar_Panel_Voltage_X
- 3.3_Bus_Current
- Receiver_Doppler
- Total_Photo_Current
```

At what time steps are sampling/learning/running stopped?

```
timesteps_stop:
  learning: 100
  running: 110
  sampling: 50
```

Is there a model for each feature, or one model combining all?

Are timestamp feature be included in models?

What's the name of timestamp feature?

What are the encoder params for timestamp?

```
models_encoders:
  minmax_percentiles:
    - 1
    - 99
  n: 700
  n_buckets: 140
  sparsity: 0.02
  timestamp:
    enable: false
    feature: satellite_time
    timeOfDay:
      - 30
      - 1
    weekend: 21
```

Is the htm.core predictor be active?

What is predictor resolution?

How many steps ahead does predictor go?

```
models_predictor:
  enable: false
  resolution: 1
  steps_ahead:
    - 1
    - 2
```

```
models_state:
  learn: true
  mode: sample_data
  model_for_each_feature: true
  timestep: 0
```

# Config Structure

## Set by user

What are the params for htm.core.AnomalyLikelihood?

What are the params for htm.core.Predictor?

What are the params for htm.core.SpatialPooler?

What are the params for htm.core.TemporalMemory?

```
models_params:
  anomaly:
    period: 1000
  predictor:
    sdrc_alpha: 0.1
  sp:
    boostStrength: 3.0
    columnCount: 1638
    localAreaDensity: 0.04395604395604396
    potentialPct: 0.85
    synPermActiveInc: 0.04
    synPermConnected: 0.13999999999999999
    synPermInactiveDec: 0.006
  tm:
    activationThreshold: 17
    cellsPerColumn: 13
    initialPerm: 0.21
    maxSegmentsPerCell: 128
    maxSynapsesPerSegment: 64
    minThreshold: 10
    newSynapseCount: 32
    permanenceDec: 0.1
    permanenceInc: 0.1
```

# Function Call Tree

source.pipeline.htm stream.stream to htm()

load\_config  
load\_json  
validate\_config

if mode == sample:  
    extend\_features\_samples

elif mode == initialize:  
    extend\_features\_samples  
    **build\_enc\_params**  
        get\_rdse\_resolution  
    **init\_models**  
        **HTMModel.init\_model()**  
            **HTMModel.init\_encs()**  
                htm.core.RDSE\_Parameters()  
                htm.core.RDSE()  
                htm.core.DateEncoder()  
            **HTMModel.init\_sp()**  
                htm.core.SpatialPooler()  
            **HTMModel.init\_tm()**  
                htm.core.TemporalMemory()  
            **HTMModel.init\_anomalyhistory()**  
                htm.core.AnomalyLikelihood()  
            **HTMModel.init\_predictor()**  
                htm.core.Predictor()  
    save\_models  
        save\_data\_as\_pickle

elif mode == run:  
    load\_models  
        load\_pickle\_object\_as\_data  
    **run\_models**  
        **HTMModel.run()**  
            **HTMModel.get\_encoding()**  
                htm.core.encoder.encode()  
                htm.core.SDR.concatenate()  
            htm.core.sp.compute()  
            **HTMModel.get\_predcount()**  
                htm.core.tm.activateDendrites()  
                htm.core.tm.getPredictiveCells()  
            htm.core.tm.compute()  
            **HTMModel.get\_preds()**  
                htm.core.predictor.infer()  
                htm.core.predictor.learn()  
    save\_outputs  
    save\_models