Detecting Silent Data Corruption Using an Auxiliary Method

And External Observer

Hadrien Croubois

École Normale Supérieure de Lyon, France

Introduction

Silent data corruption (SDC) is corruption of the memory which:

- Is not detected by hardware level fault tolerance mechanisms,
- Does not cause major disruption of the systems' execution,
- Does not cause major disruption of the applications' execution.

Silent data corruption (SDC) is corruption of the memory which:

Issue

The risk is that those errors might stay unnoticed and threaten the results' validity.

- Does not cause major disruption of the applications' execution.

There are two sorts of SDC:

Systematic SDC

SDC that are expected to affect all executions of the application in a similar way

(ex: bugs)

Nonsystematic SDC

SDC happening randomly

(ex: radiation induced bitflips)

Process replication

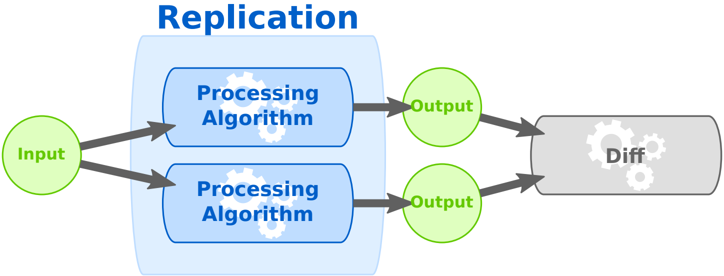
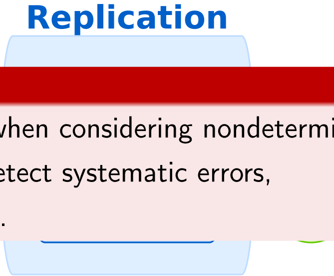


Figure 1: Simple pipeline replication

Process replication



Limitation

- Difficult when considering nondeterministic algorithms,
- Cannot detect systematic errors,
- Expensive.

Figure 1: Simple pipeline replication

Predictive detection

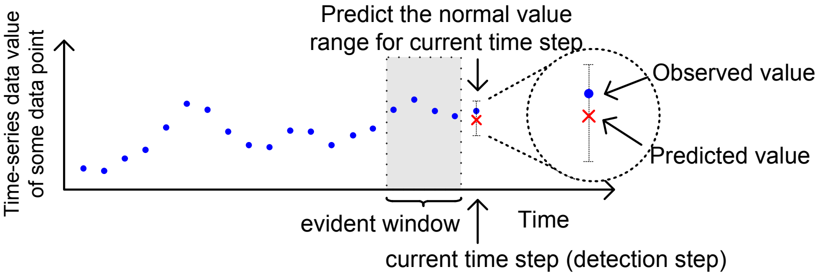
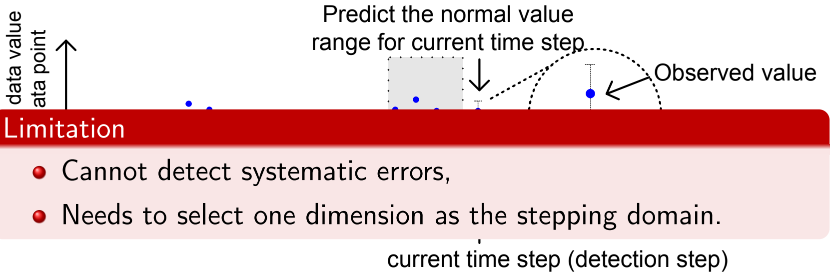


Figure 2: Predictive detection

Di & al. An Efficient Silent Data Corruption Detection Method with Error-Feedback Control and Even Sampling for HPC Applications. 2015.

Predictive detection



Limitation

- Cannot detect systematic errors,
- Needs to select one dimension as the stepping domain.

Figure 2: Predictive detection

Di & al. An Efficient Silent Data Corruption Detection Method with Error-Feedback Control and Even Sampling for HPC Applications. 2015.

A new detection mechanism

Our objective is to provide a new SDC detection mechanism:

- With low overhead,
- Capable of detecting both systematic & nonsystematic SDC,
- Applicable to acyclic pipelines (no temporal dimension),
- Deployed at the workflow level.

Figure 3: Variable replication with external observer

Multialgorithms replication pipeline

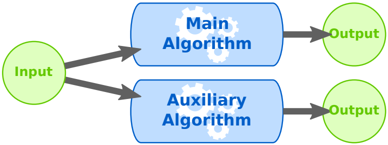


Figure 3: Variable replication with external observer

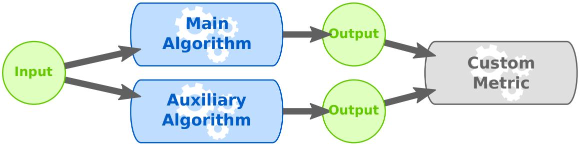


Figure 3: Variable replication with external observer

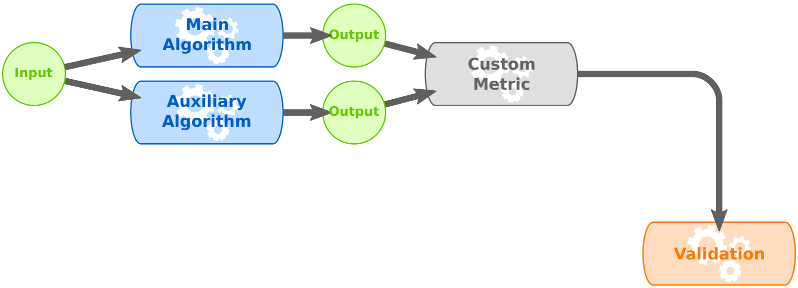


Figure 3: Variable replication with external observer

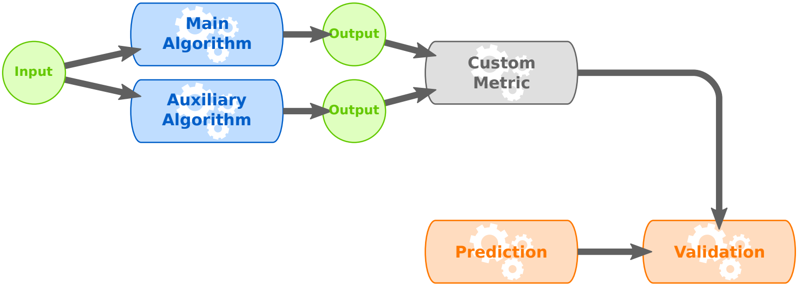


Figure 3: Variable replication with external observer

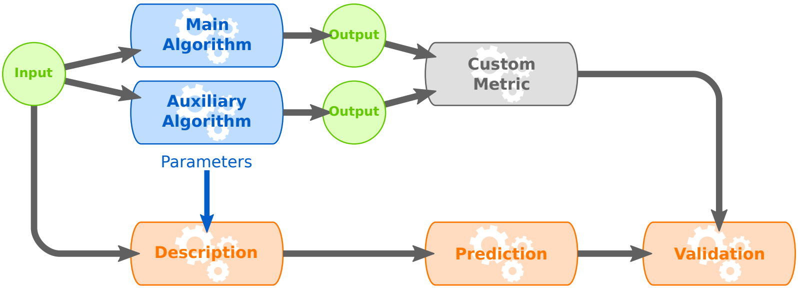


Figure 3: Variable replication with external observer

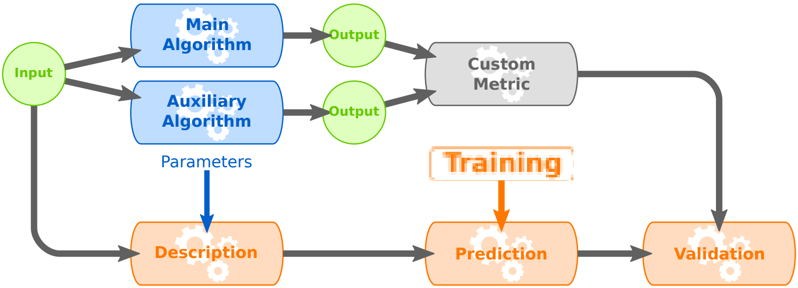


Figure 3: Variable replication with external observer

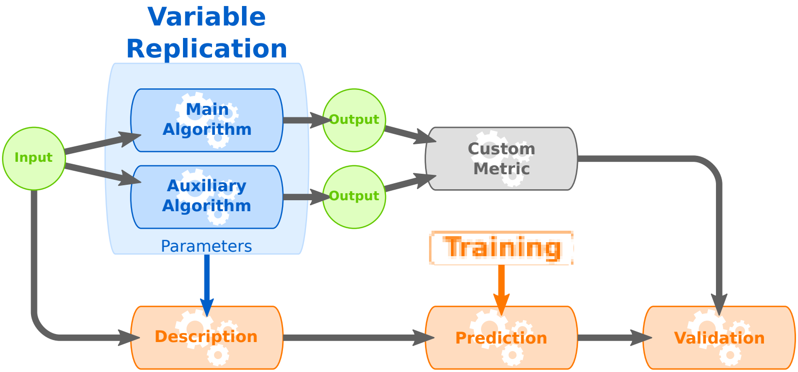


Figure 3: Variable replication with external observer

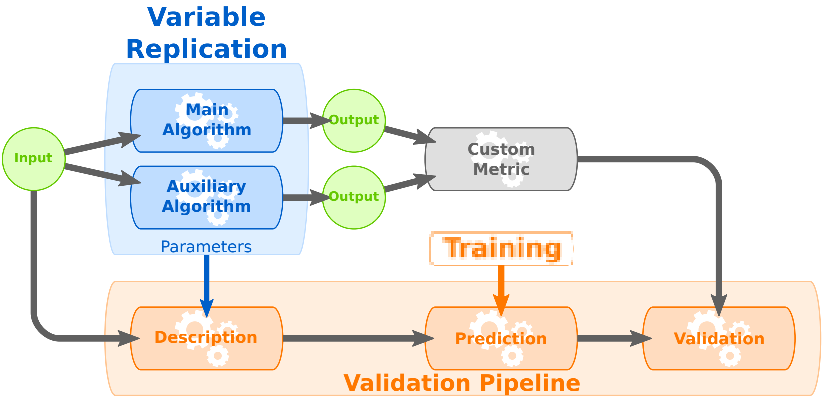


Figure 3: Variable replication with external observer

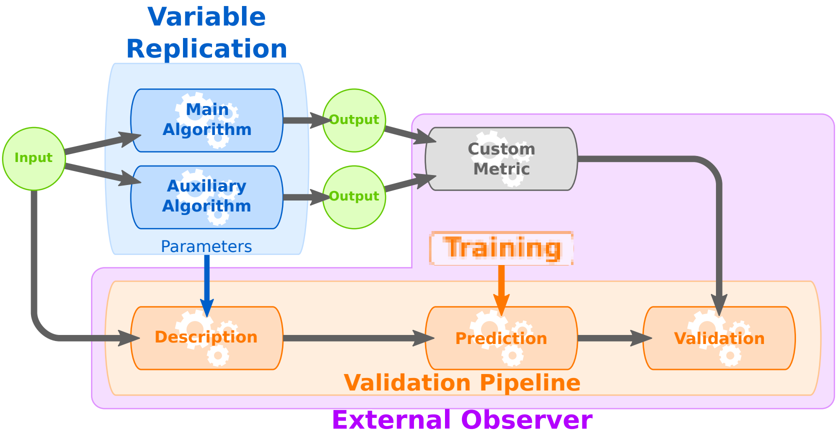


Figure 3: Variable replication with external observer

Applying this model to a use case :

Ptychography reconstruction

What is Ptychography ?

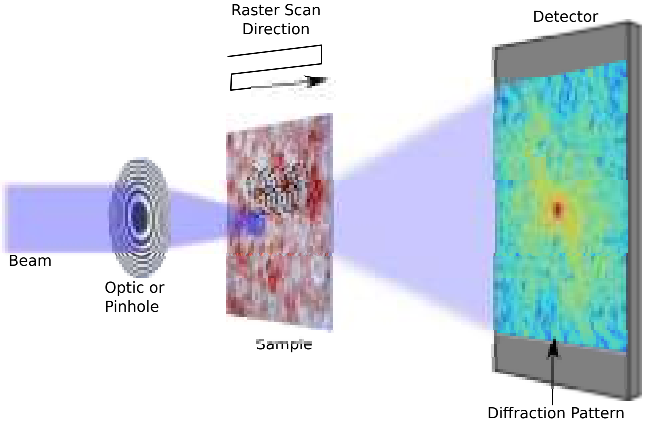


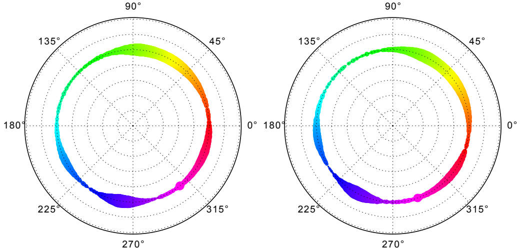
Figure 4: Simplified ptychography setup

Nashed & al. Parallel ptychographic reconstruction 2014

Building a metric

Reconstruction injects artifacts that have to be removed :

1. Phase shift correction,

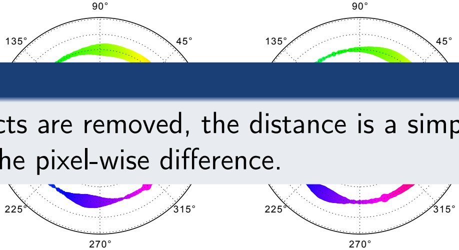


2. Phase gradient correction.

Building a metric

Reconstruction injects artifacts that have to be removed :

1. Phase shift correction,



Limitation

Once those artifacts are removed, the distance is a simple RMS measurement of the pixel-wise difference.

2. Phase gradient correction.

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,
2. (Inserting corruption in the diffraction patterns),

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,
2. (Inserting corruption in the diffraction patterns),
3. Reconstructing the transmission characteristics,

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,
2. (Inserting corruption in the diffraction patterns),
3. Reconstructing the transmission characteristics,
4. Comparing the results (distance evaluation),

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,
2. (Inserting corruption in the diffraction patterns),
3. Reconstructing the transmission characteristics,
4. Comparing the results (distance evaluation),
5. Validating the measured distance.

Experimental protocol

For testing the detection capability we deploy a full pipeline with purposely injected SDC :

1. Generating simulated diffraction patterns,
2. (Inserting corruption in the diffraction patterns),
3. Reconstructing the transmission characteristics,
4. Comparing the results (distance evaluation),
5. Validating the measured distance.

We also will be evaluating the trade-off between sensitivity and cost when changing the number of observed diffraction patterns.

Results 1/3

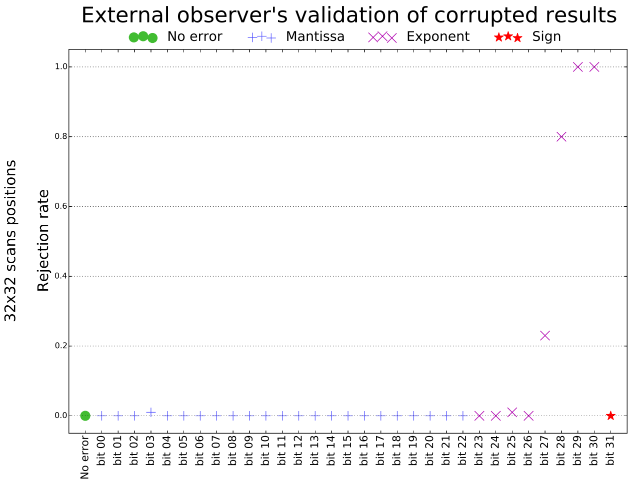


Figure 5: Acceptance rate depending on the position of injected errors

Results 2/3

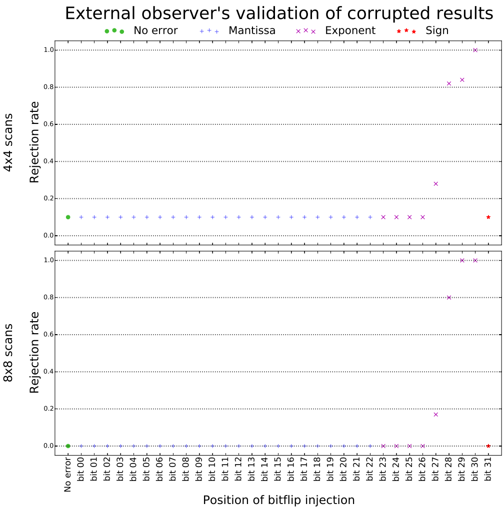


Figure 6: Acceptance rate depending on the position of injected errors with reduced complexity auxiliary methods

Results 3/3

Auxiliary algorithm's scan dimension	False positive (fp)	Detection rate (d)
4×4	10.00%	8.82%
8×8	0.00%	9.28%
32×32	0.00%	9.53%

Table 1: False positive and detection rates for our ptychography replicated pipeline validating corrupted reconstruction results.

$$fp = rejection(\text{no errors})$$

$$d = \frac{1}{32} \sum_{b \in [0,31]} \frac{rejection(b) - fp}{1 - fp}$$

Bootstrapping the prediction model

Detection is made possible by the prediction model.

Detection is made possible by the prediction model.

Prediction model is built through training.

Detection is made possible by the prediction model.

Prediction model is built through training.

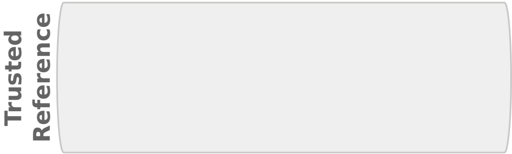
How to handle SDC during this training phase ?

Detection is made possible by the prediction model.

Challenge

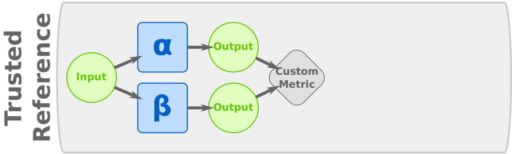
We need a way build a trustworthy prediction model.

How to handle SDC during this training phase ?



Involved algorithms:

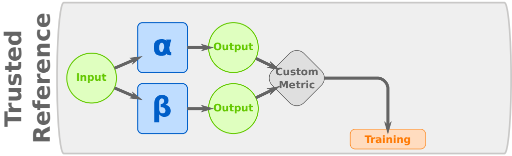
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),

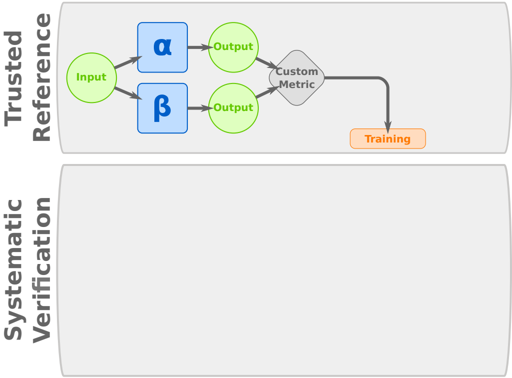
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),

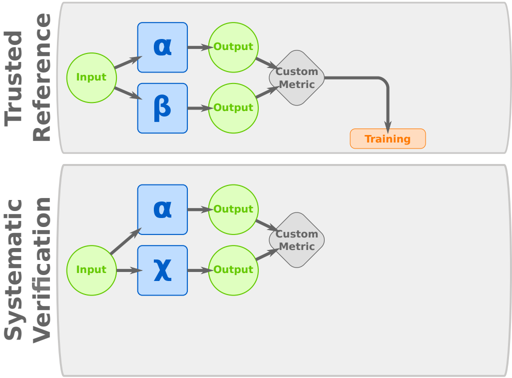
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),

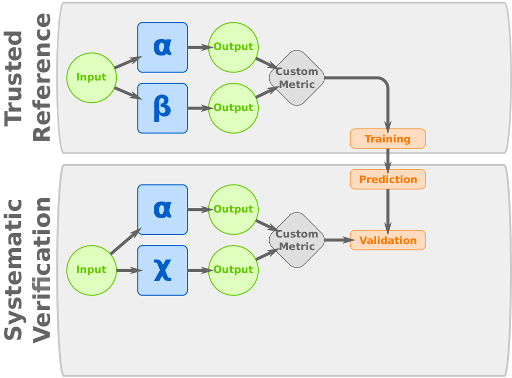
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),

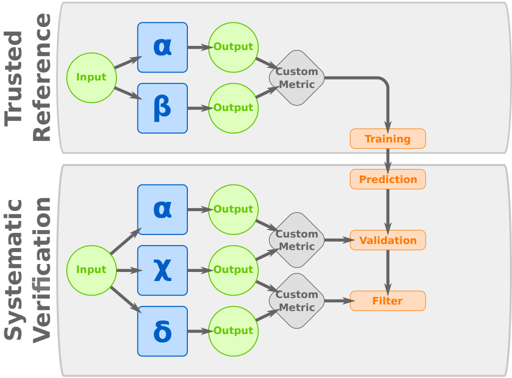
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),

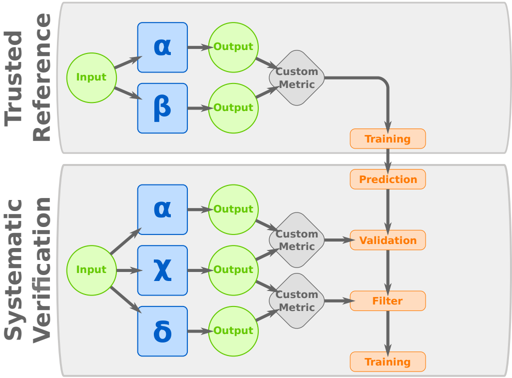
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

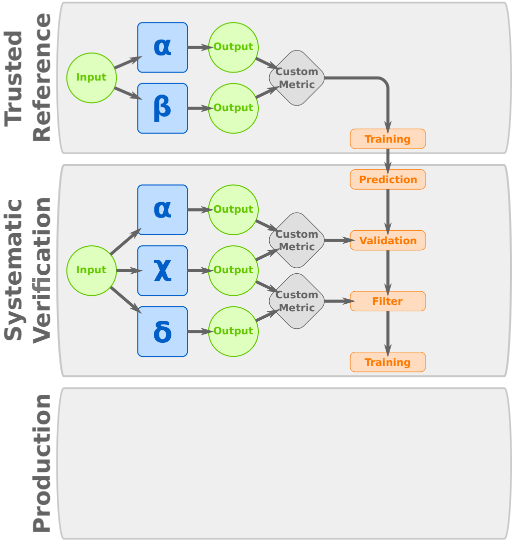
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

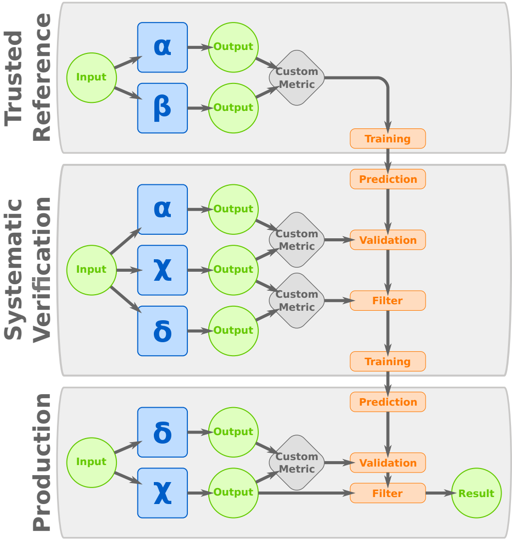
Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

Figure 7: Bootstrapping a prediction model for systematic SDC detection

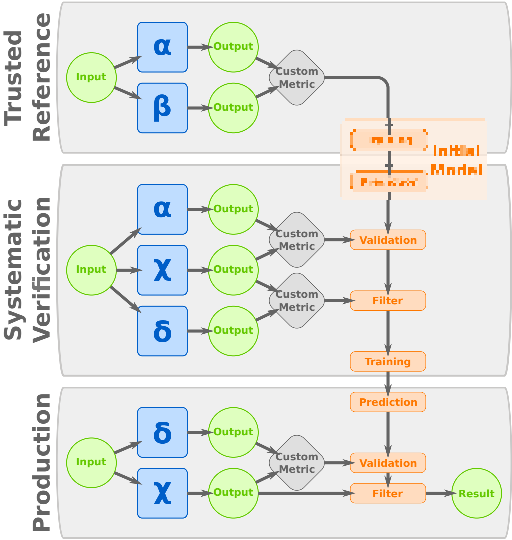


Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

Figure 7: Bootstrapping a prediction model for systematic SDC detection

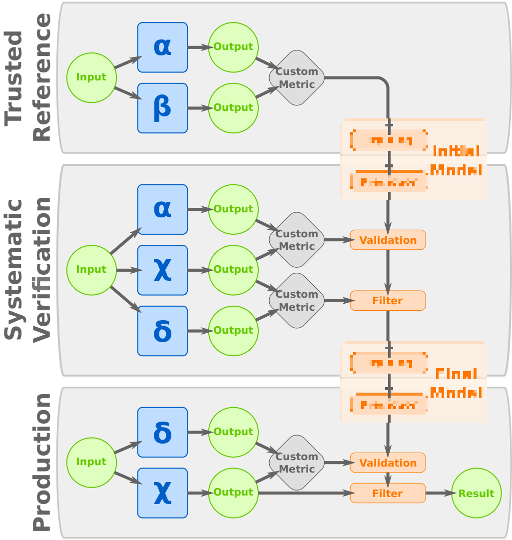
The bootstrapping pipeline



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

Figure 7: Bootstrapping a prediction model for systematic SDC detection



Involved algorithms:

- α Expensive reference (truth),
- β Previously used algorithm (balanced),
- χ New algorithm (to be proven better then β),
- δ Auxiliary algorithm for the production pipeline.

Figure 7: Bootstrapping a prediction model for systematic SDC detection

Applying this model to an HPC use case :

Density estimation

Density estimation

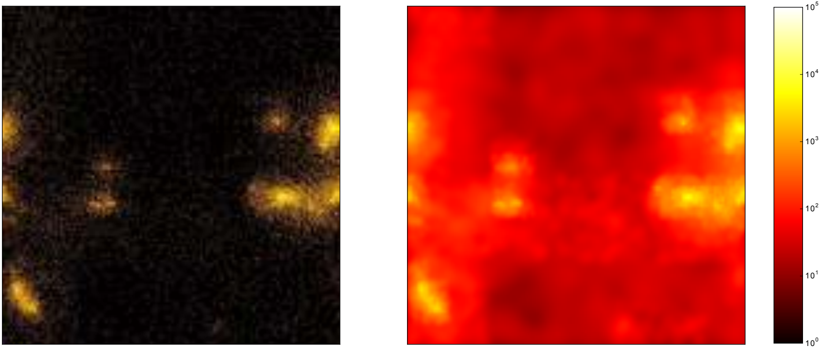
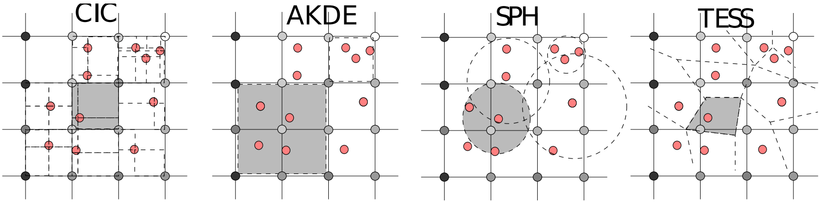


Figure 8: From a set of particles to a density field: visualisation of dark matter distribution in the early universe

Algorithms description



$\mathcal{O}(p + r^3)$ $\mathcal{O}(p \log p + r^3)$ $\mathcal{O}(t.p \log p + r^3)$ $\mathcal{O}(p^2 + r^3)$

Figure 9: Weight distribution mechanisms of different density estimators

Peterka & al. Self-adaptive density estimation of particle data 2015

Natural resiliency

Those algorithms are naturally resilient to noise in the input data.

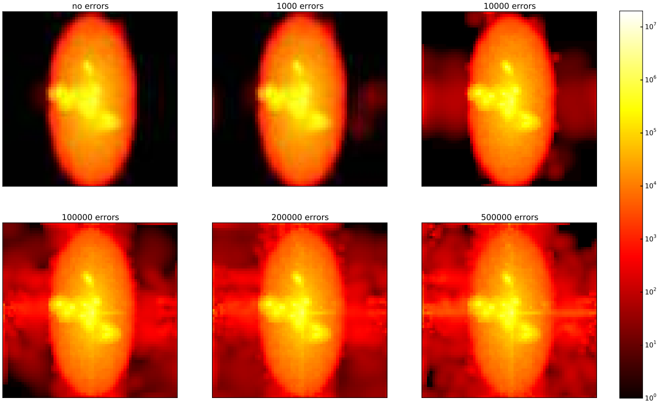


Figure 10: Impact of random bitflips on a 200k particles distribution.
$1000 = 0.005\%$, $10000 = 0.05\%$, $100000 = 0.5\%$

Custom metric

To reflect the structure of produced density fields, we build a custom metric based on the radial powerspectrum.

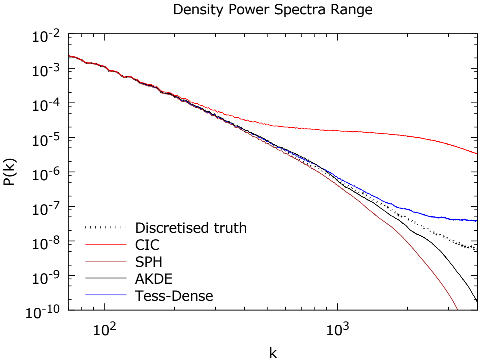
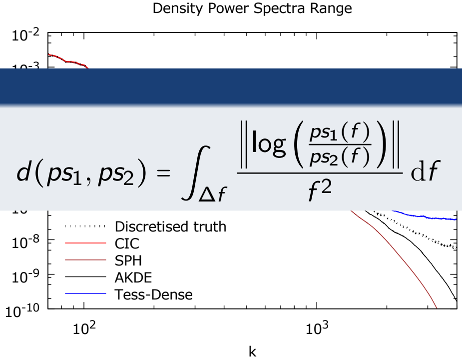


Figure 11: Radial powerspectrum response of different density estimators

Custom metric

To reflect the structure of produced density fields, we build a custom metric based on the radial powerspectrum.



Distance

$$d(ps_1, ps_2) = \int_{\Delta f} \frac{\left\| \log\left(\frac{ps_1(f)}{ps_2(f)}\right) \right\|}{f^2} \, \mathrm{d}f$$

Figure 11: Radial powerspectrum response of different density estimators

Experimental protocol 1/2

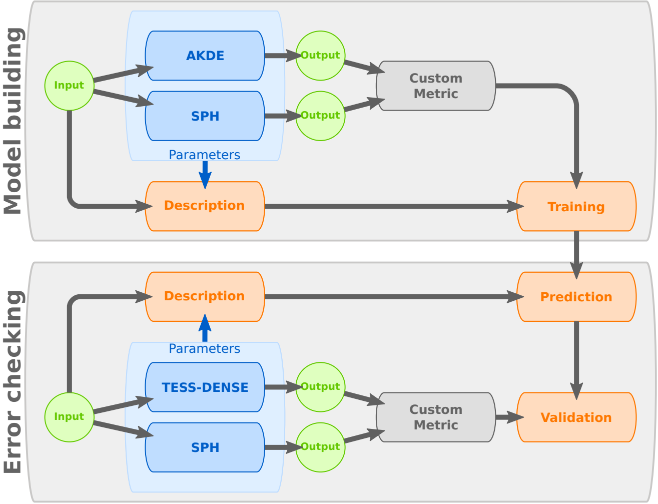


Figure 12: Deploying the bootstrapping pipeline for systematic SDC detection

Experimental protocol 2/2

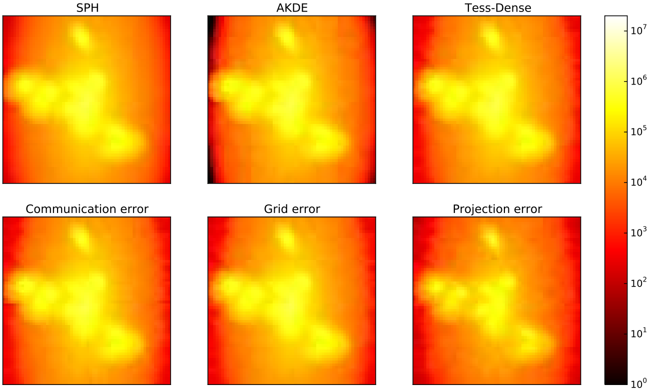


Figure 13: Density field computed by different density estimators. Top: SPH, AKDE, Tess-Dense. Bottom: Tess-Dense with communication error, Tess- Dense with grid error, Tess-Dense with projection error

Results

Build	Bug type	Acceptance rate
Clean	None	100.00%
Communication error	Systematic	3.75%
Grid error	Systematic	0.00%
Projection error	Systematic	8.75%
Projection datarace	Probabilistic	58.75%

Table 2: Acceptance rates of the bootstrapping process for different versions of Tess-Dense

Conclusion

Capabilities

- Fully generic replication mechanism,
- Provide protection against both systematic and nonsystematic SDC,
- Low overhead in the production pipeline,
- Validation based on results meaningfulness ratter then exact matching,
- Users can specify the characteristic of the results they are interested in.

Limitation

- Requires an (affordable) auxiliary algorithm,
- Relies on machine learning,
- Provides detection but no correction.

Future Work

- Comparative study of machine learning processes,
- Implementation as generic model in Decaf.

Conclusion

Thank you for your attention.

Any questions ?