Dependability Evaluation and Enhancing Methods for Transformer Models

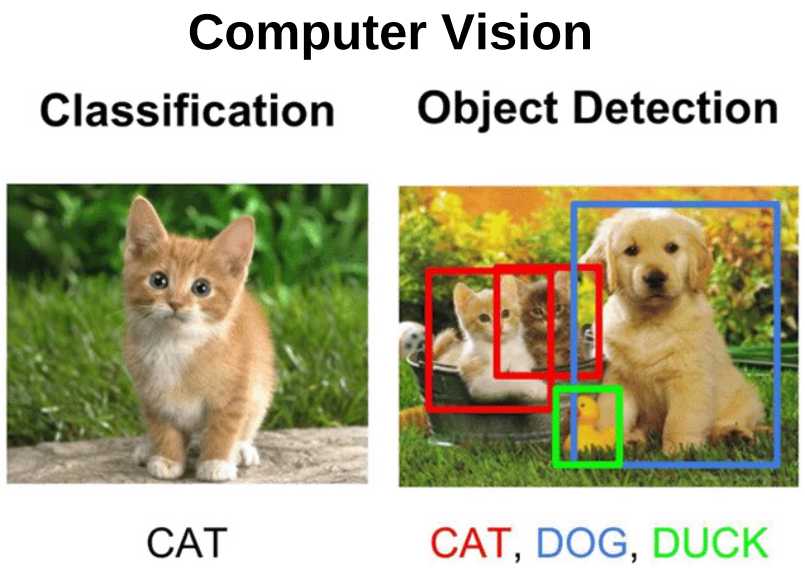
ETS PHD FORUM

Transformers models

- Transformers, state-of-the-art of machine learning (ML) models
  - Highly accurate, very complex
  - Many tasks

Transformers models

- Transformers, state-of-the-art of machine learning (ML) models
  - Highly accurate, very complex
  - Many tasks



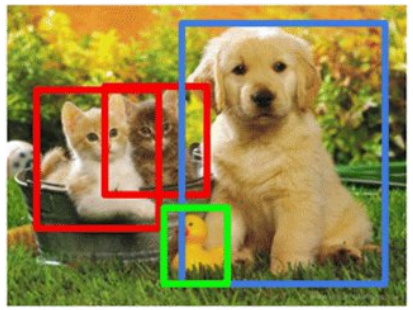
Transformers models

- Transformers, state-of-the-art of machine learning (ML) models
  - Highly accurate, very complex
  - Many tasks

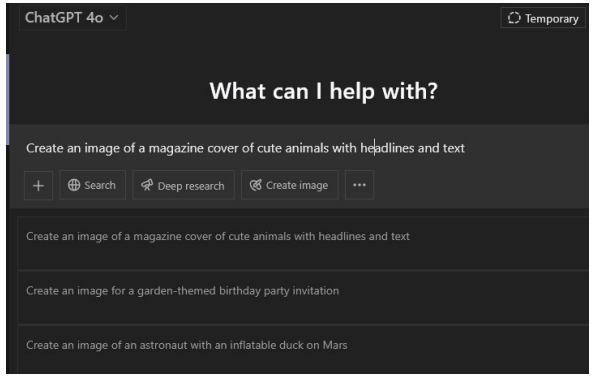
Computer Vision **NLP**

Classification **Object Detection**

CAT



CAT, DOG, DUCK



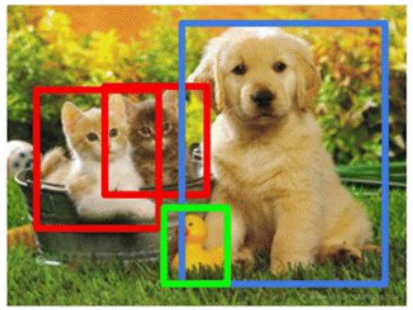
Transformers models

- Transformers, state-of-the-art of machine learning (ML) models
  - Highly accurate, very complex
  - Many tasks

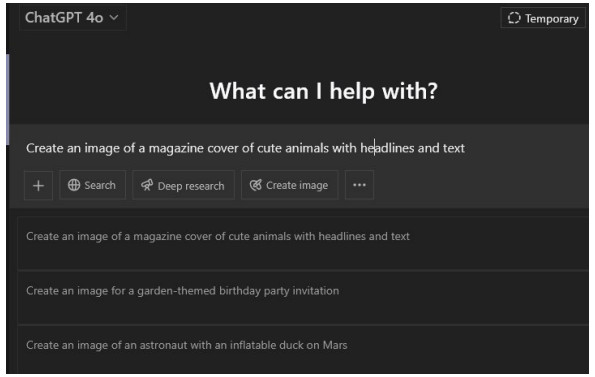
Computer Vision **NLP**

Classification **Object Detection**

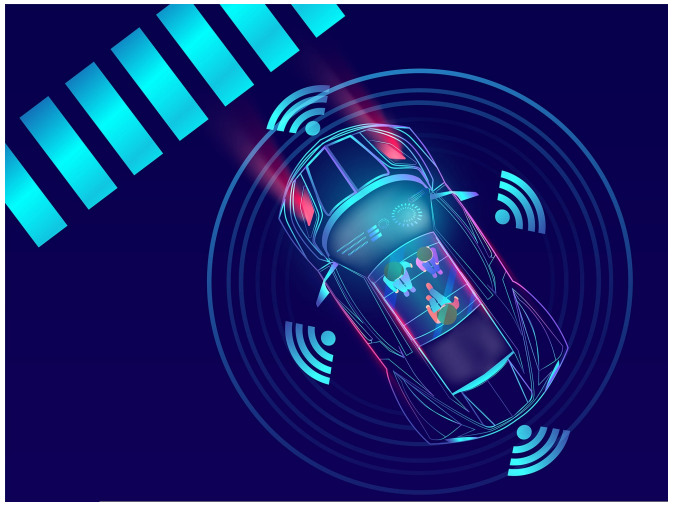
CAT



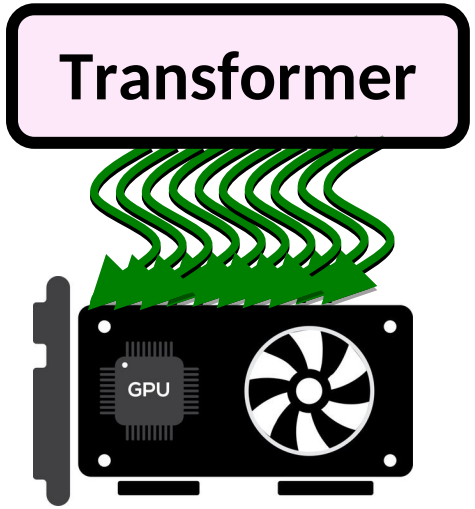
CAT, DOG, DUCK



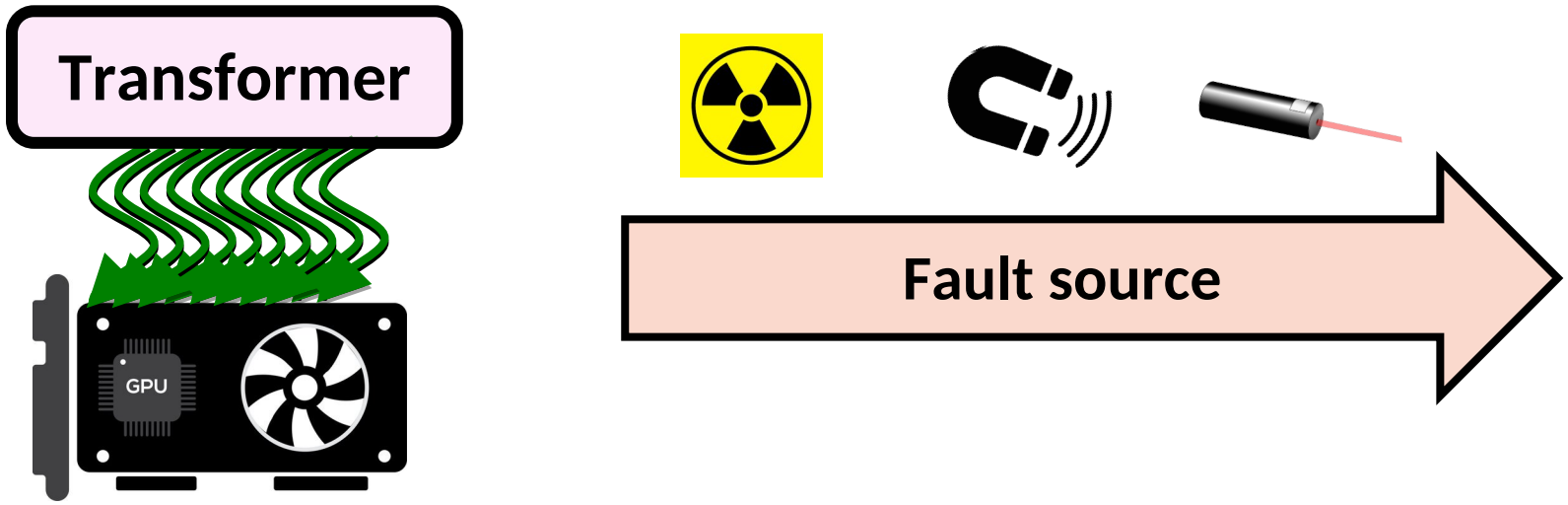
Safety-critical systems



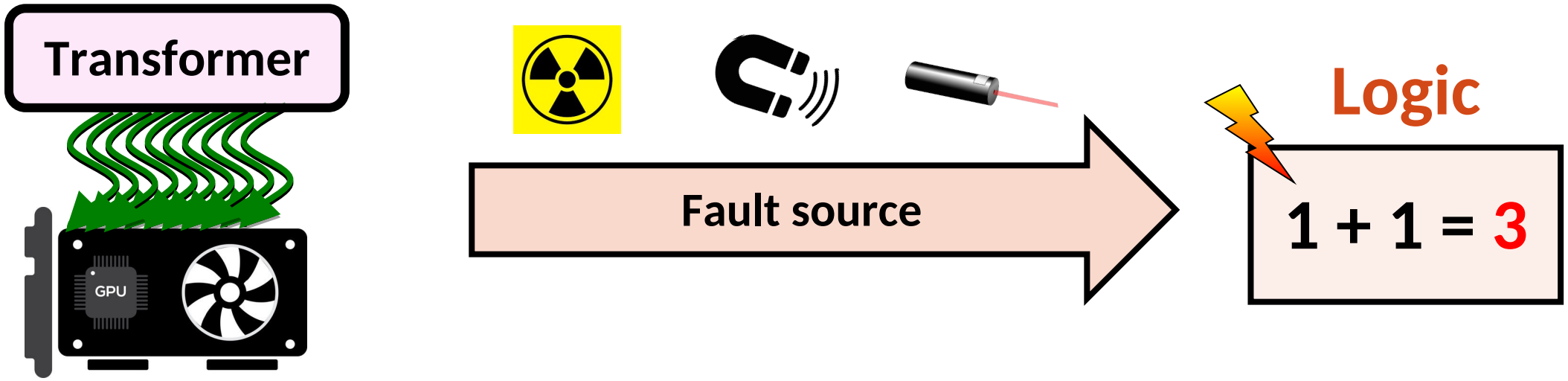
Hardware is vulnerable to faults



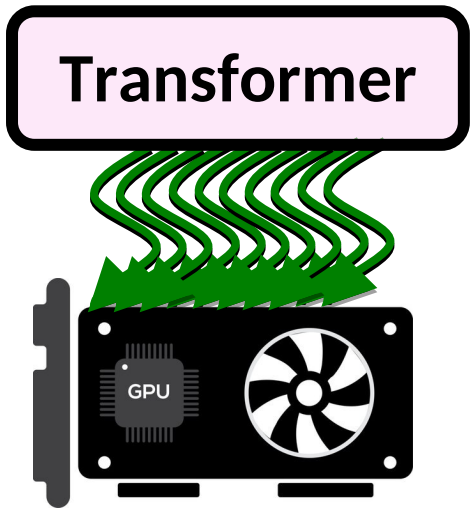
Hardware is vulnerable to faults

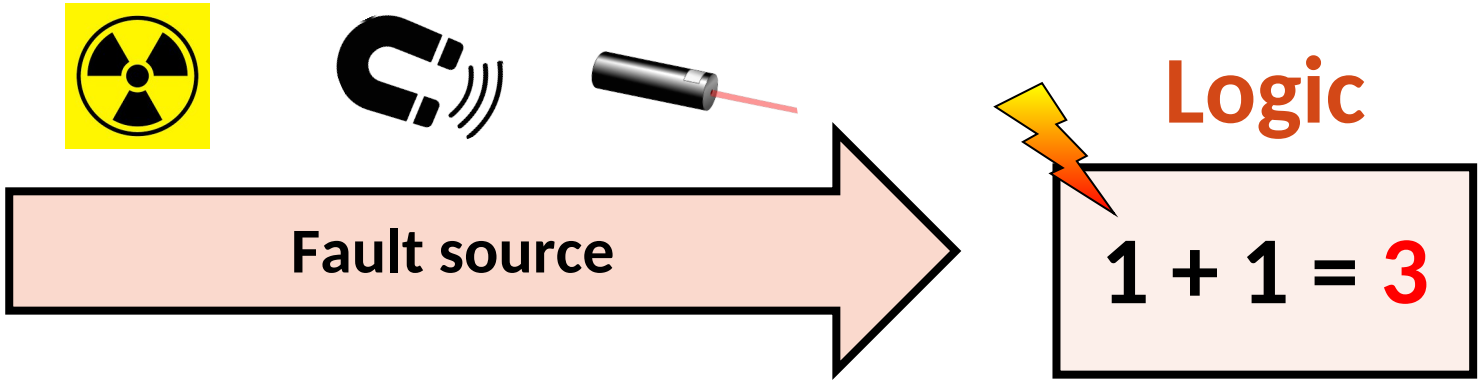


Hardware is vulnerable to faults

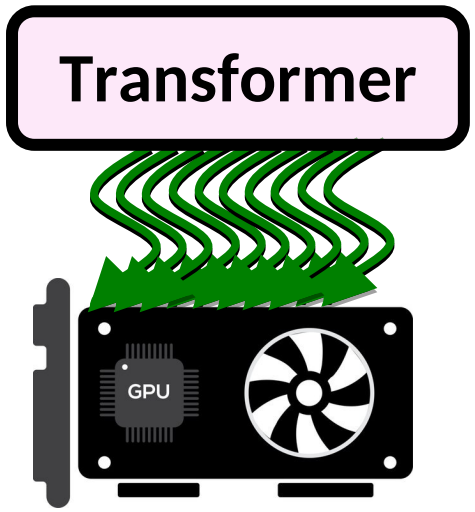


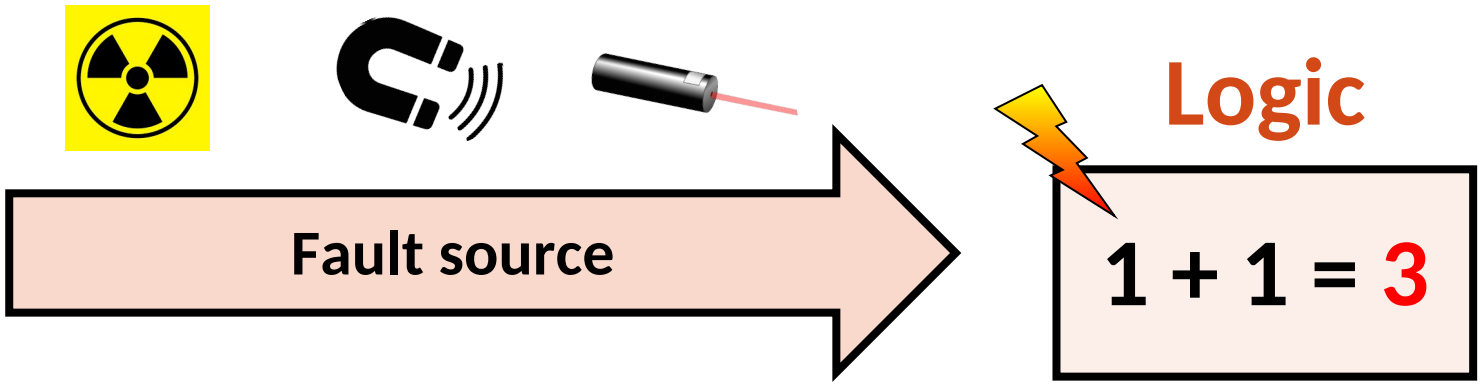
Hardware is vulnerable to faults

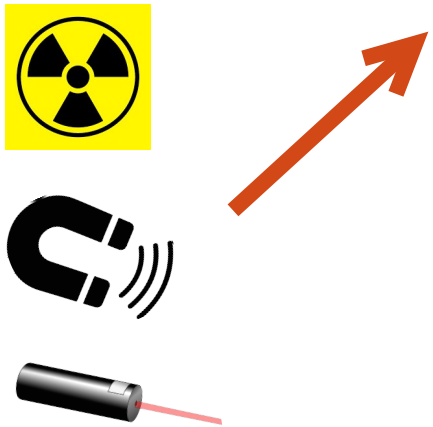




Hardware is vulnerable to faults



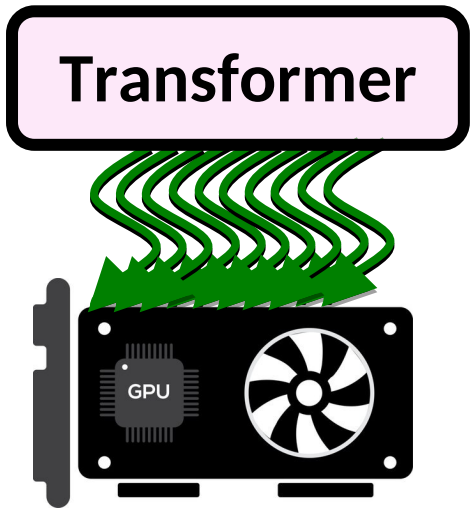


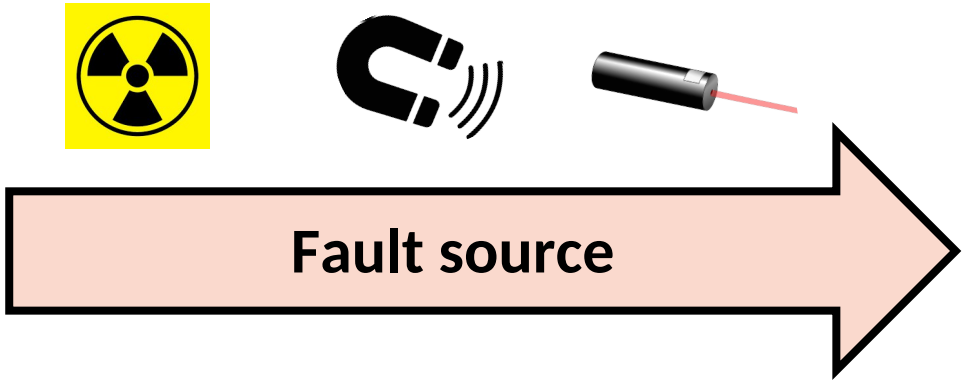


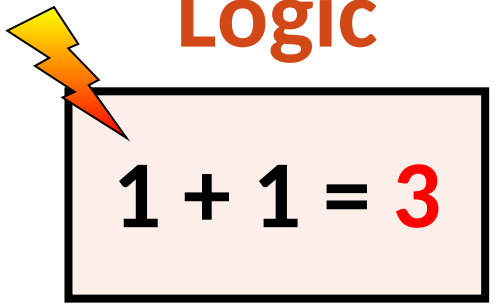
Silent Data Corruption (SDC)

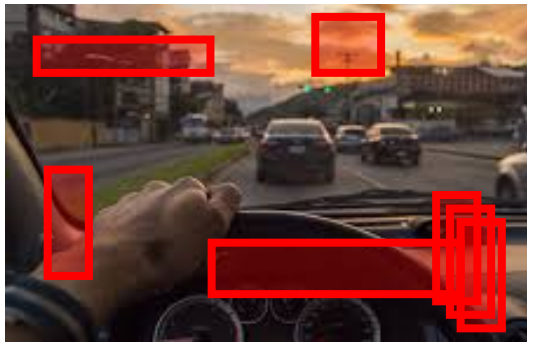
- **Computation is done but result is altered**

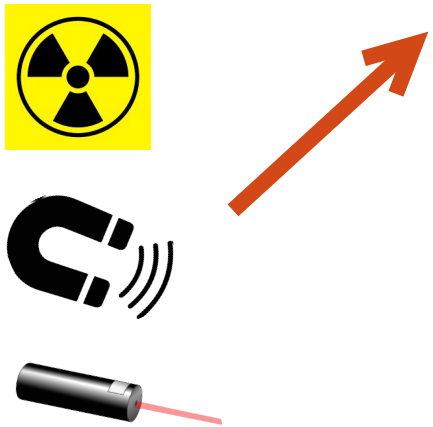
Hardware is vulnerable to faults







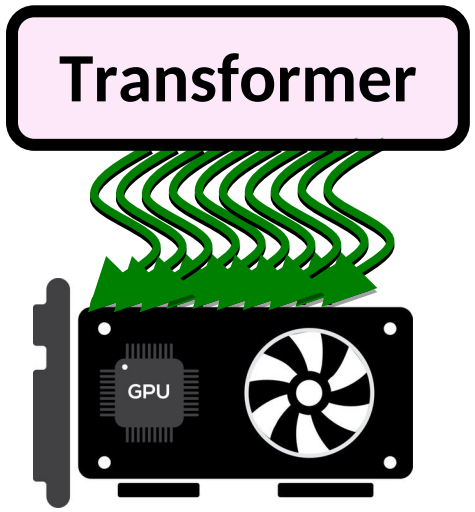


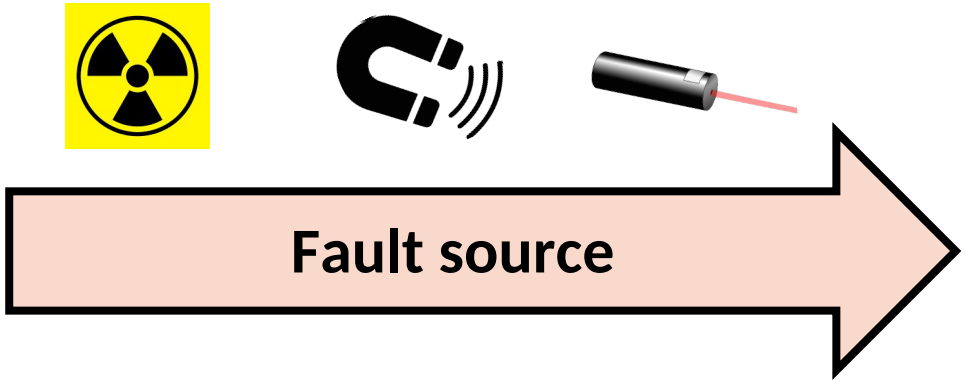


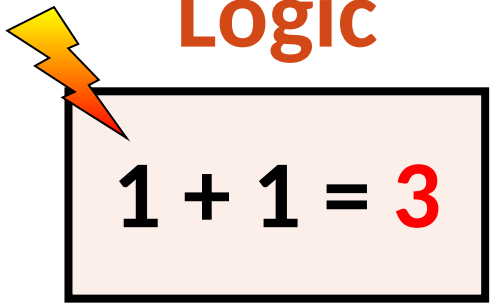
Silent Data Corruption (SDC)

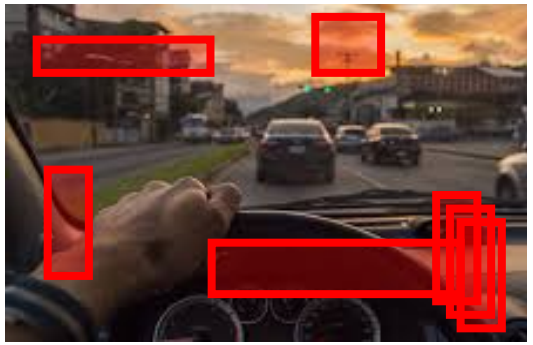
- **Computation is done but result is altered**

Hardware is vulnerable to faults









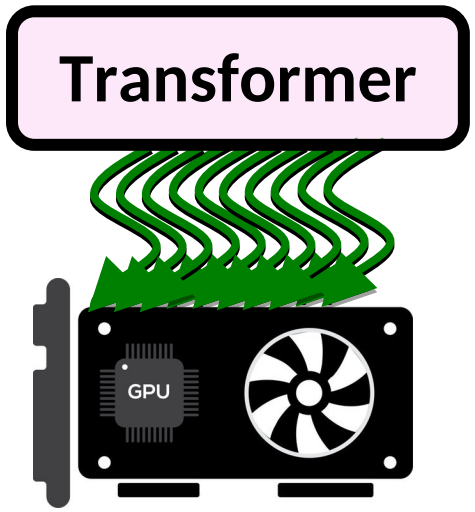
Silent Data Corruption (SDC)

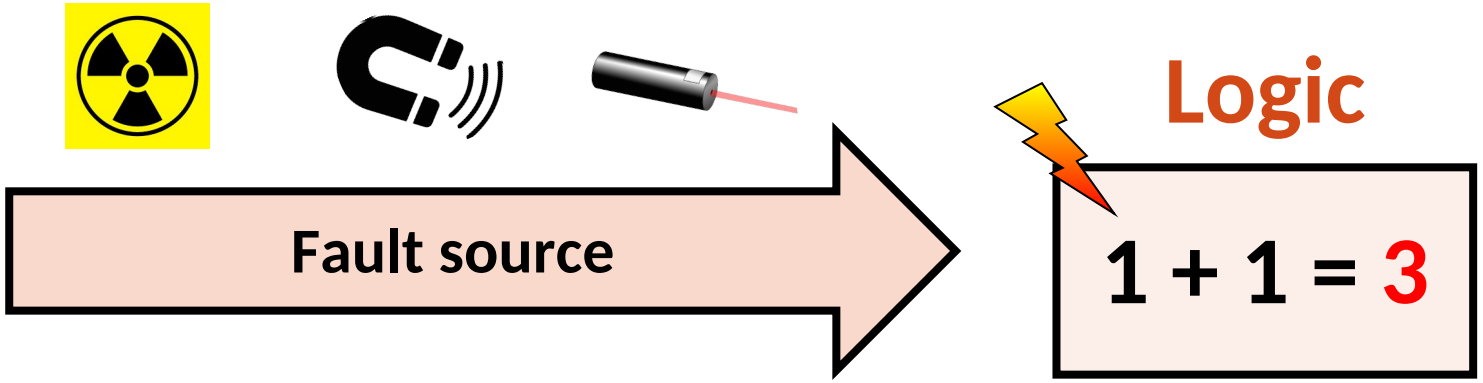
- Computation is done but result is altered

Detected Unrecoverable Error (DUE)

- Crash and OS hangs

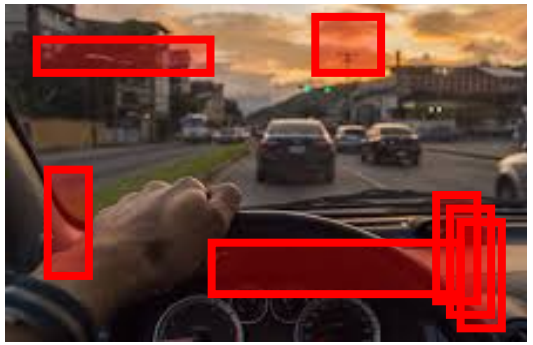
Hardware is vulnerable to faults





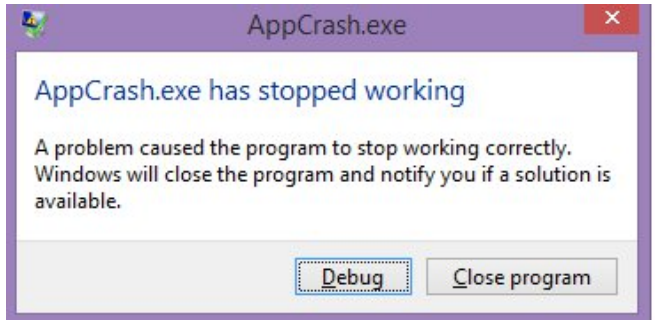
Silent Data Corruption (SDC)

- Computation is done but result is altered

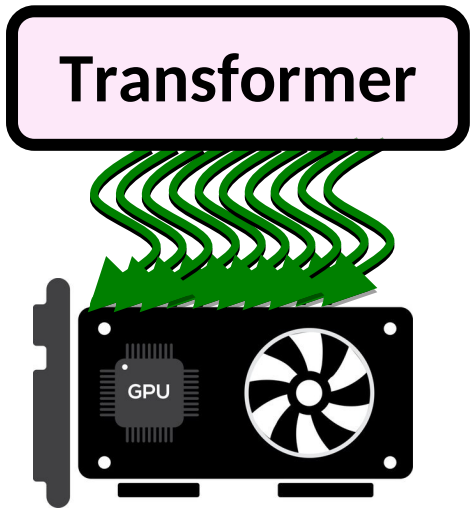


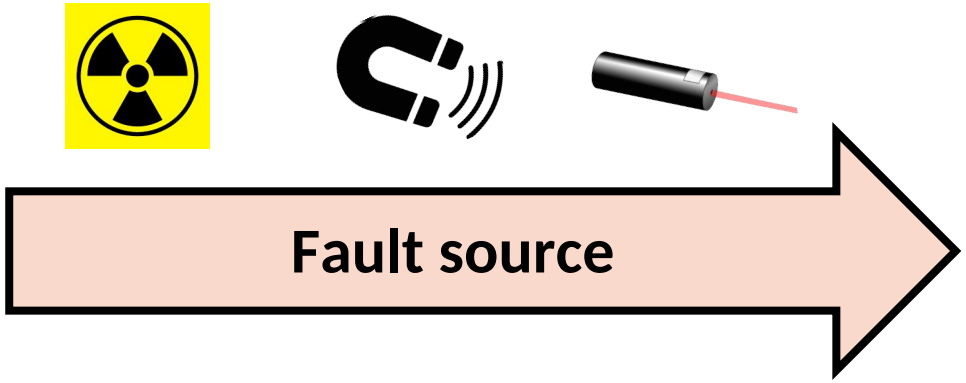
Detected Unrecoverable Error (DUE)

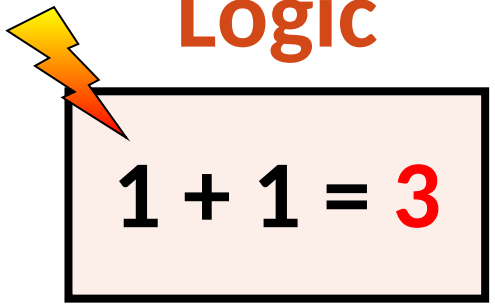
- Crash and OS hangs

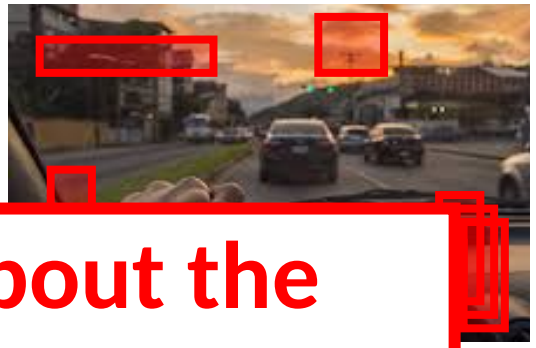


Hardware is vulnerable to faults





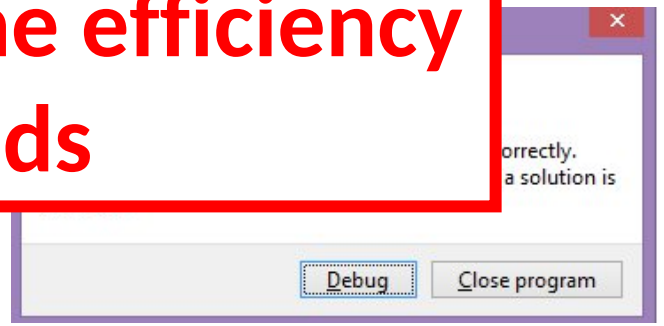




Silent Data Corruption (SDC)

This raises multiple questions about the assessement methodology and the efficiency of the protection methods

- Crash and OS hangs



One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:

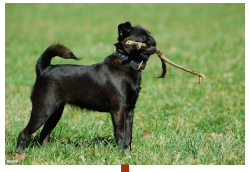
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:



Transformer

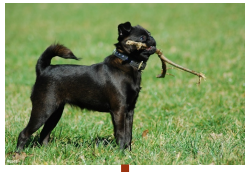
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:



Transformer

0.98	0.02	0.005	...	0.00..

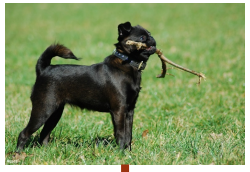
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:



Transformer

0.98	0.02	0.005	...	0.00..
Dog	Cat	Fish		Car

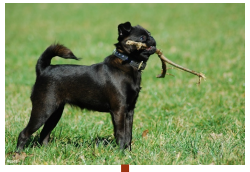
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:



Transformer

0.98	0.02	0.005	...	0.00..
Dog	Cat	Fish		Car

We define the confidence as:

$$C = Top1 - Top2$$

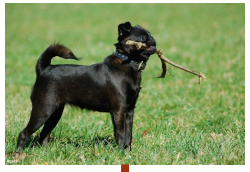
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:

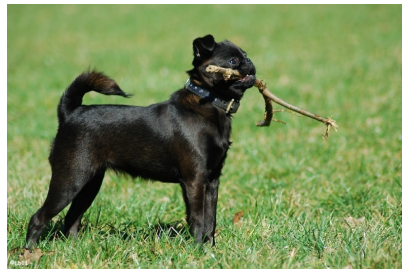


Transformer

0.98	0.02	0.005	...	0.00..
Dog	Cat	Fish		Car

We define the confidence as:

$$C = Top1 - Top2$$



Conf.: 96%, Class: Dog

Conf.: 0.1%, Class: Ambulance

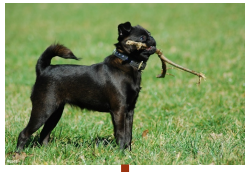
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:

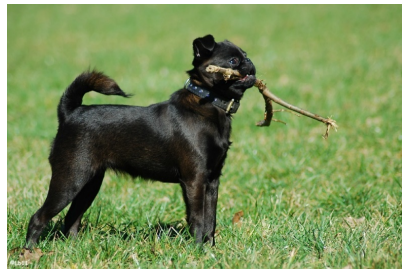


Transformer

0.98	0.02	0.005	...	0.00..
Dog	Cat	Fish		Car

We define the confidence as:

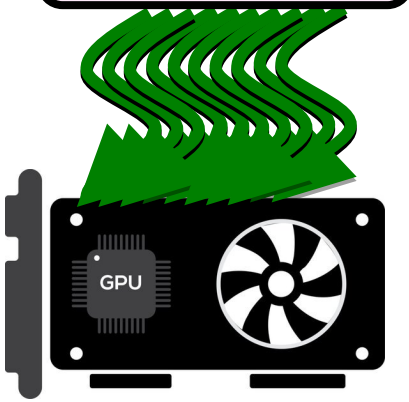
$$C = Top1 - Top2$$



Conf.: 96%, Class: Dog

Conf.: 0.1%, Class: Ambulance

Transformer



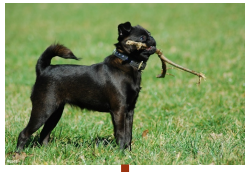
One under-explored variable: inputs

For a dataset of N classes

$Top1$	$Top2$	$Top3$	...	$TopN$

Result: class associated to $Top1$

Example:

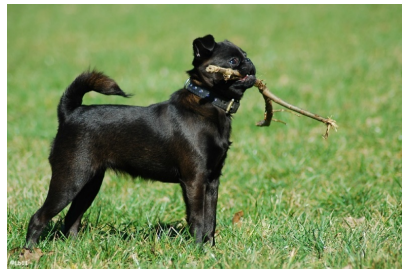


Transformer

0.98	0.02	0.005	...	0.00..
Dog	Cat	Fish		Car

We define the confidence as:

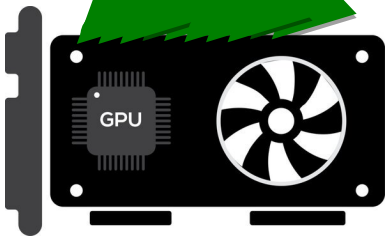
$$C = Top1 - Top2$$



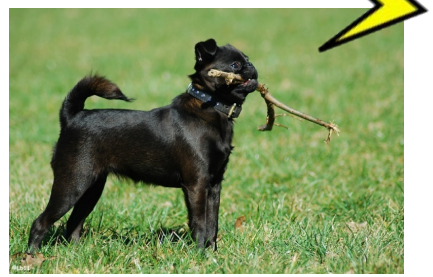
Conf.: 96%, Class: Dog

Conf.: 0.1%, Class: Ambulance

Transformer



Less critical?



Conf.: 80%, Class: Dog

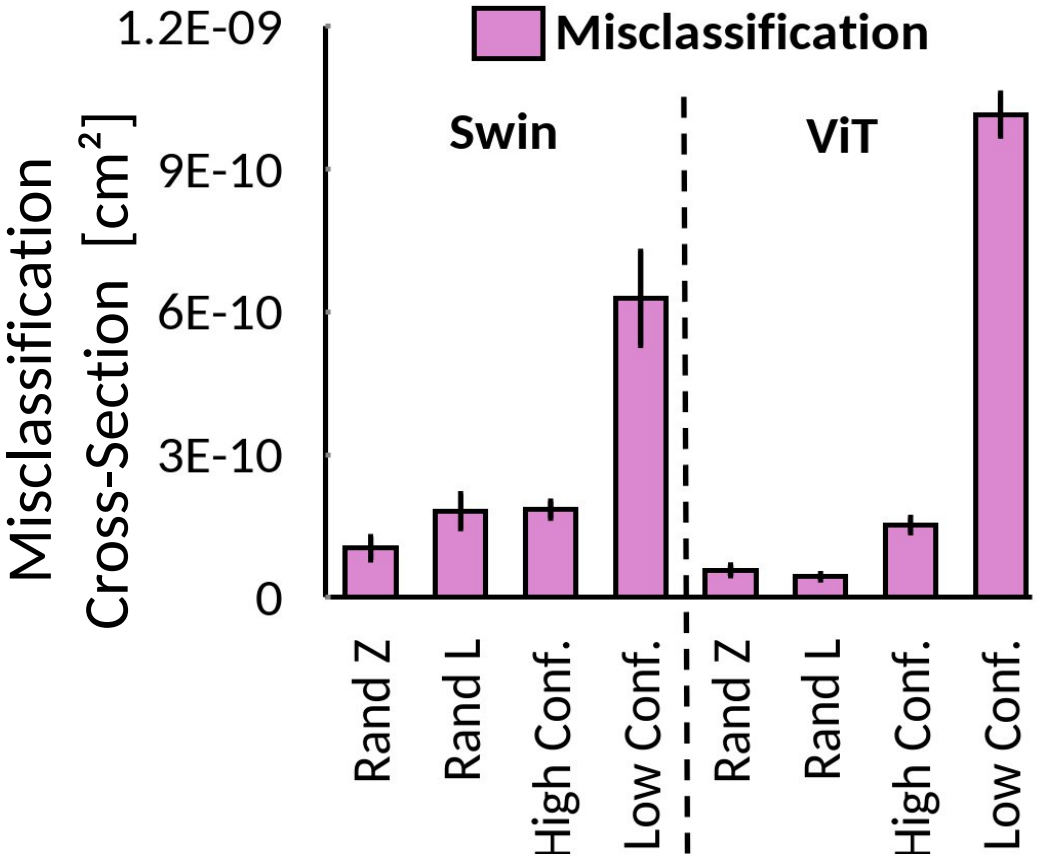
More critical?

Conf.: 5%, Class: Dog

Results in a nutshell

Results in a nutshell

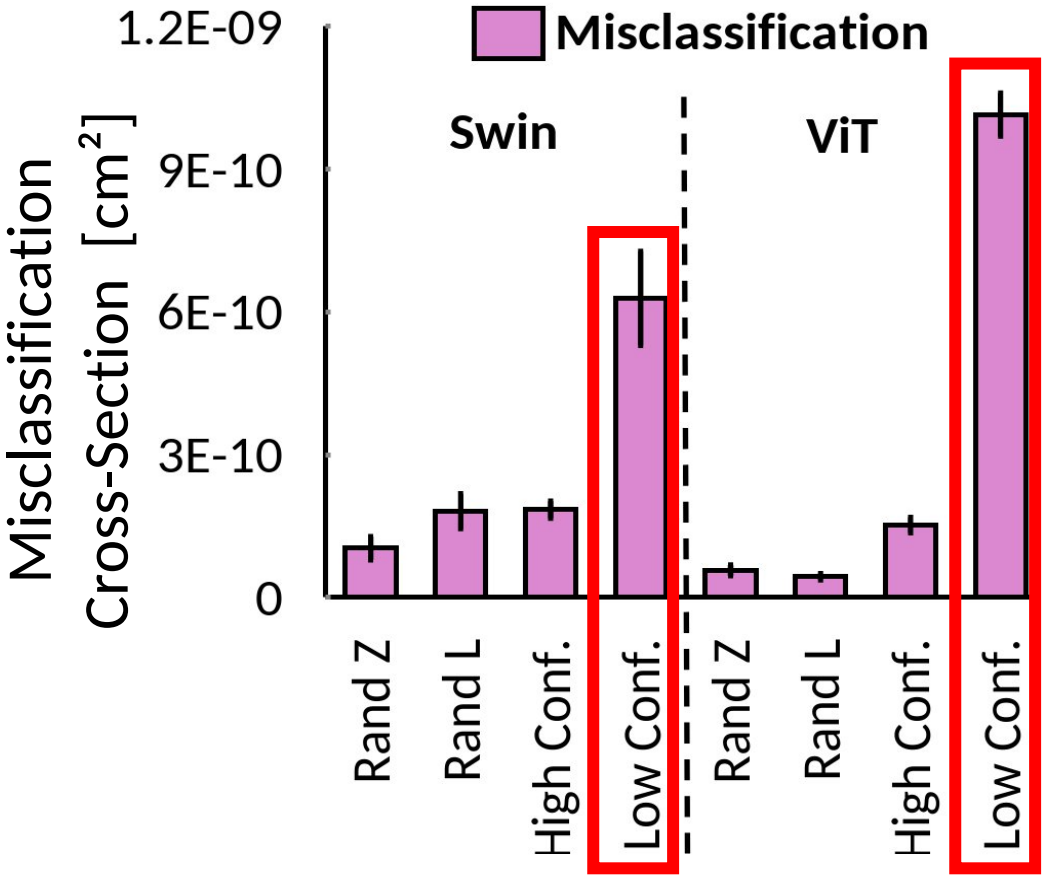
Neutron beam (unprotected) [1]*



[1] L. Roquet, IEEE TNS, 2026

Results in a nutshell

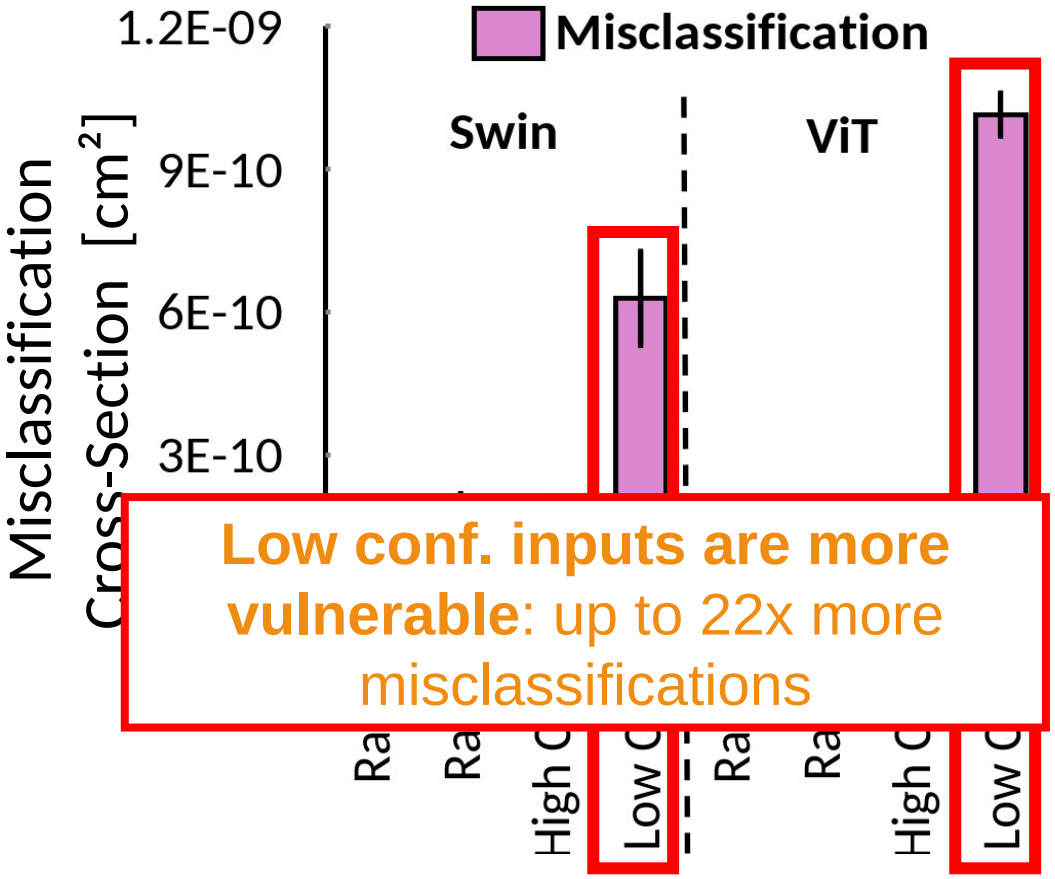
Neutron beam (unprotected) [1]*



[1] L. Roquet, IEEE TNS, 2026

Results in a nutshell

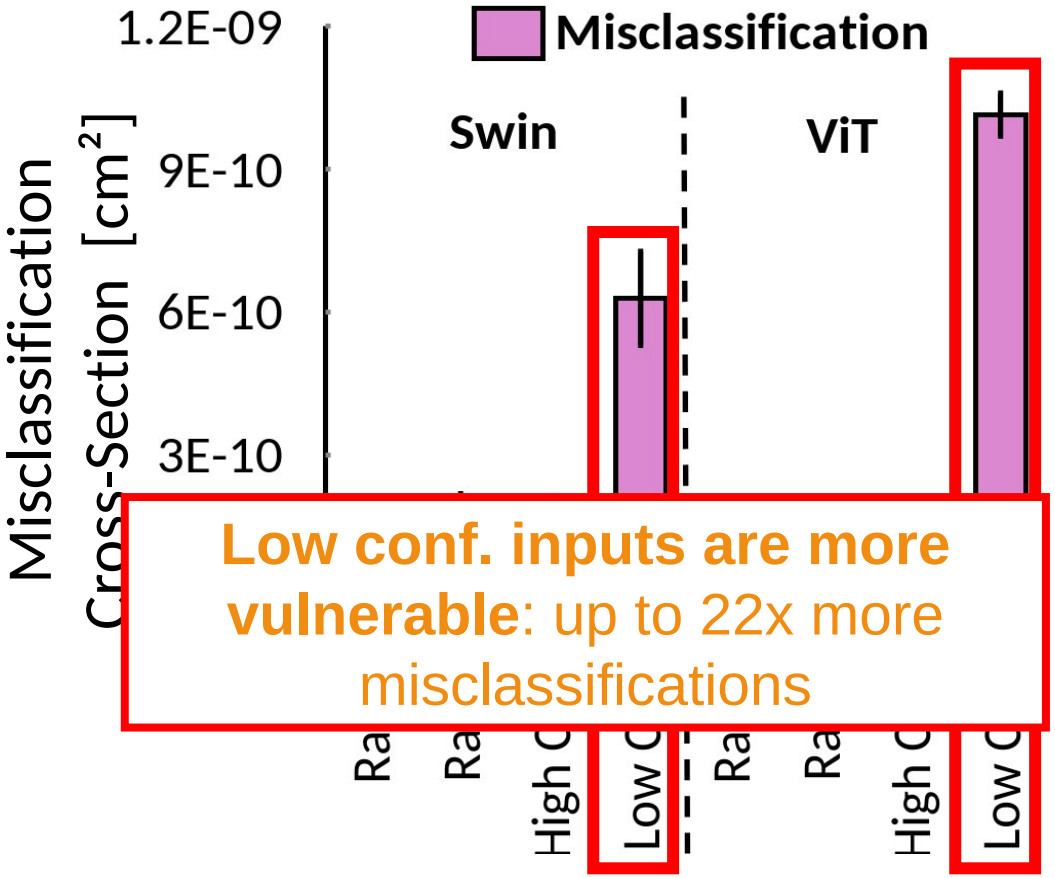
Neutron beam (unprotected) [1]*



[1] L. Roquet, IEEE TNS, 2026

Results in a nutshell

Neutron beam (unprotected) [1]*

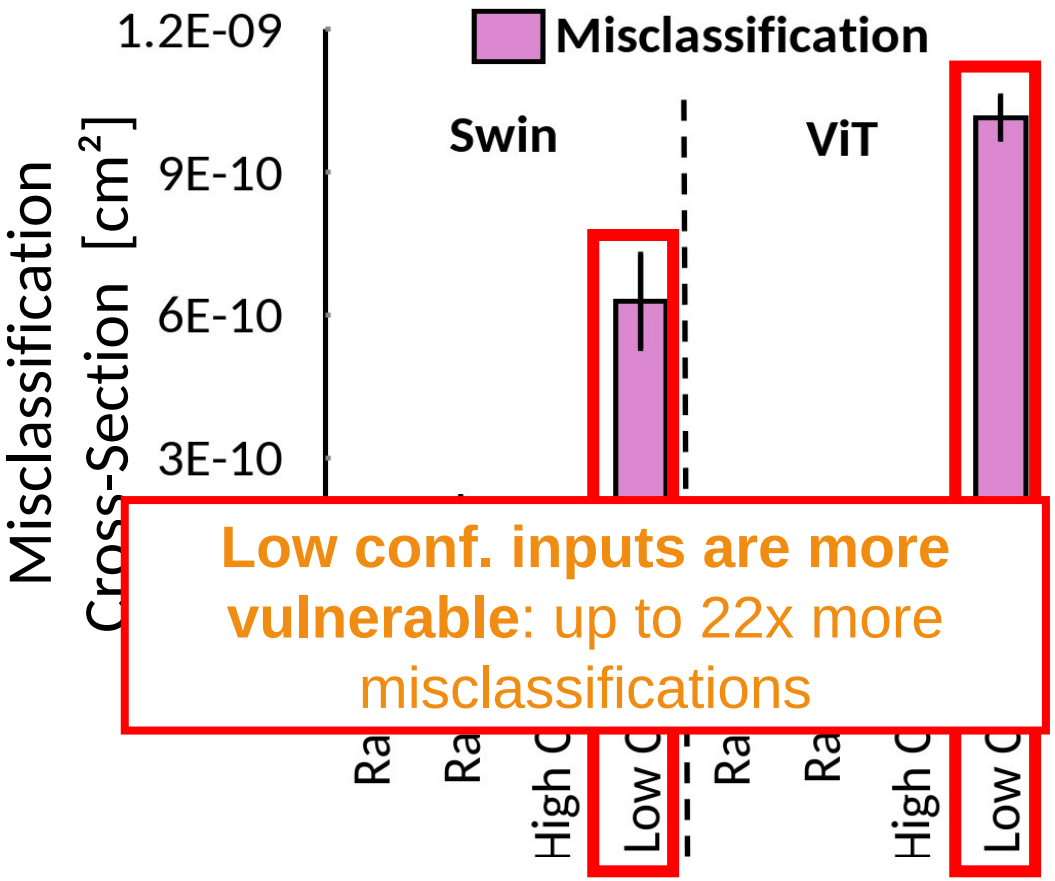


*Same trends are observed by injecting fault at software level

[1] L. Roquet, IEEE TNS, 2026

Results in a nutshell

Neutron beam (unprotected) [1]*

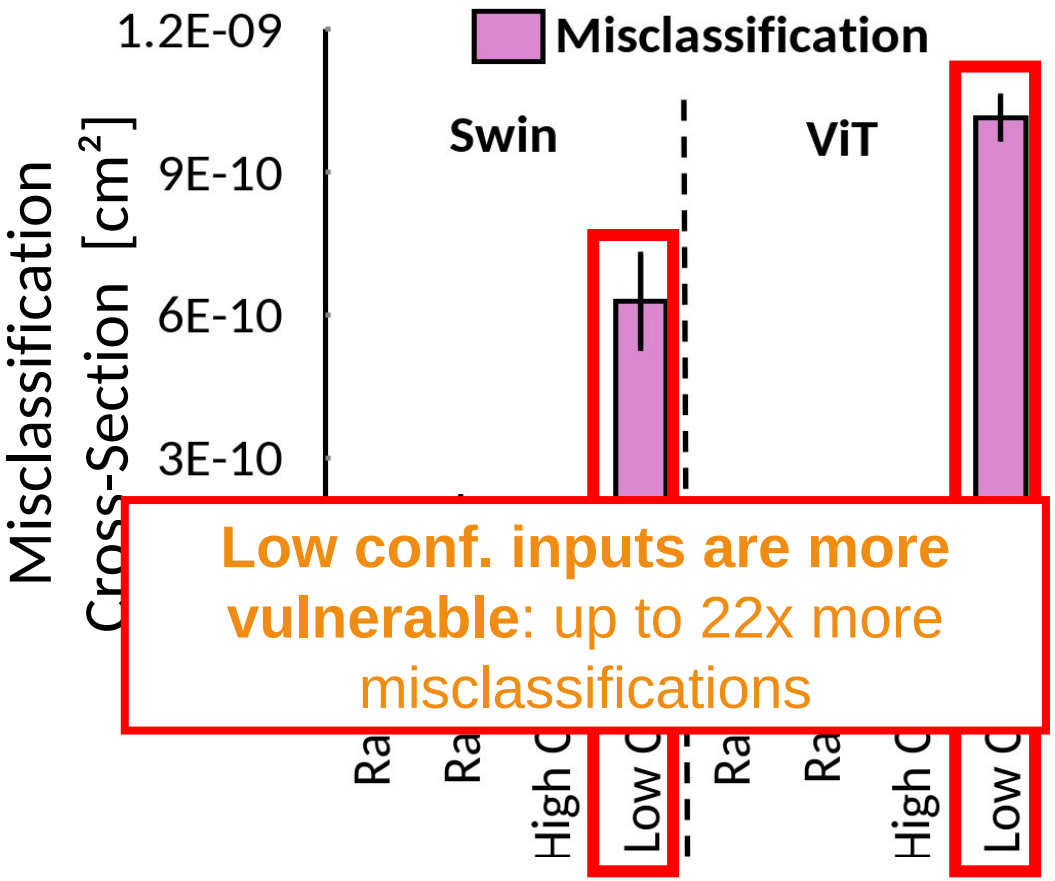


*Same trends are observed by injecting fault at software level

[1] L. Roquet, IEEE TNS, 2026

Results in a nutshell

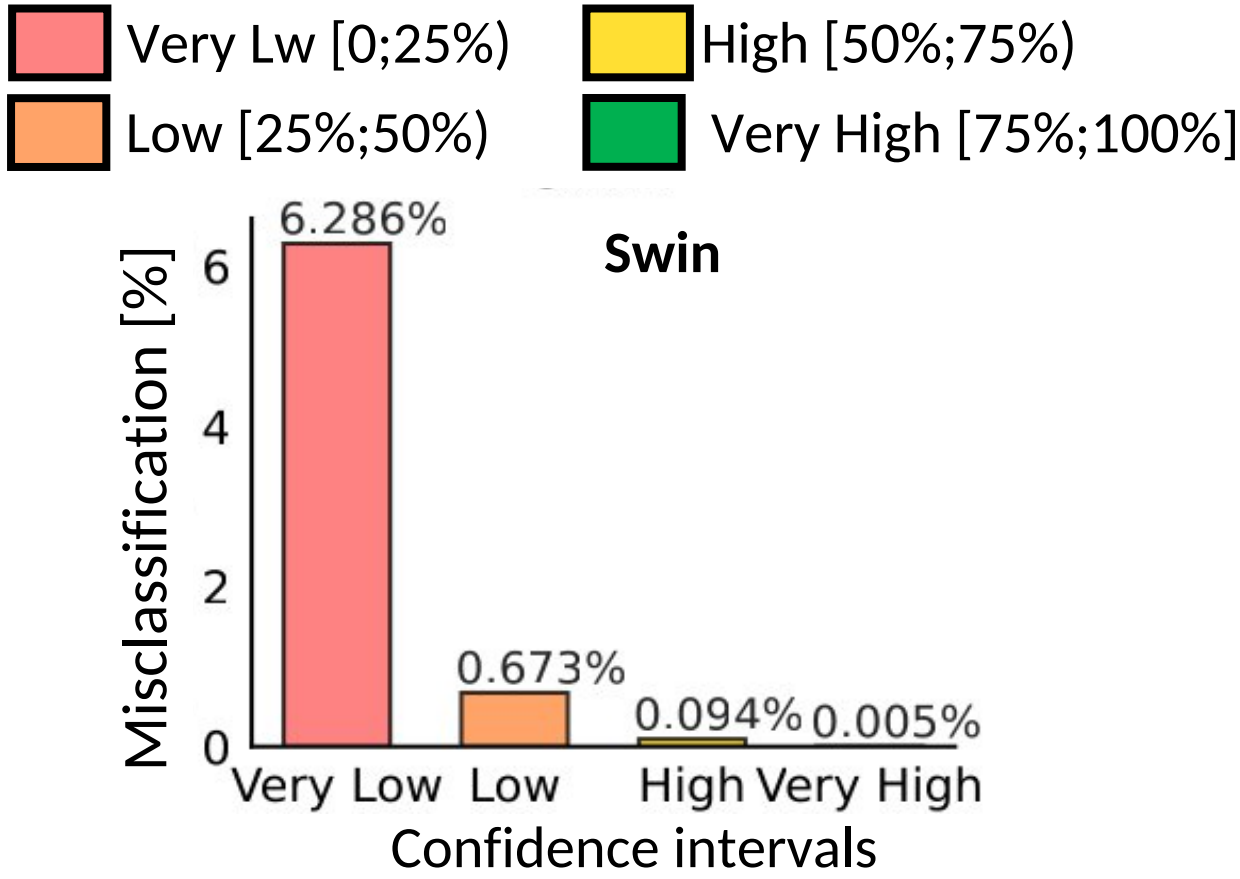
Neutron beam (unprotected) [1]*



Low conf. inputs are more vulnerable: up to 22x more misclassifications

*Same trends are observed by injecting fault at software level

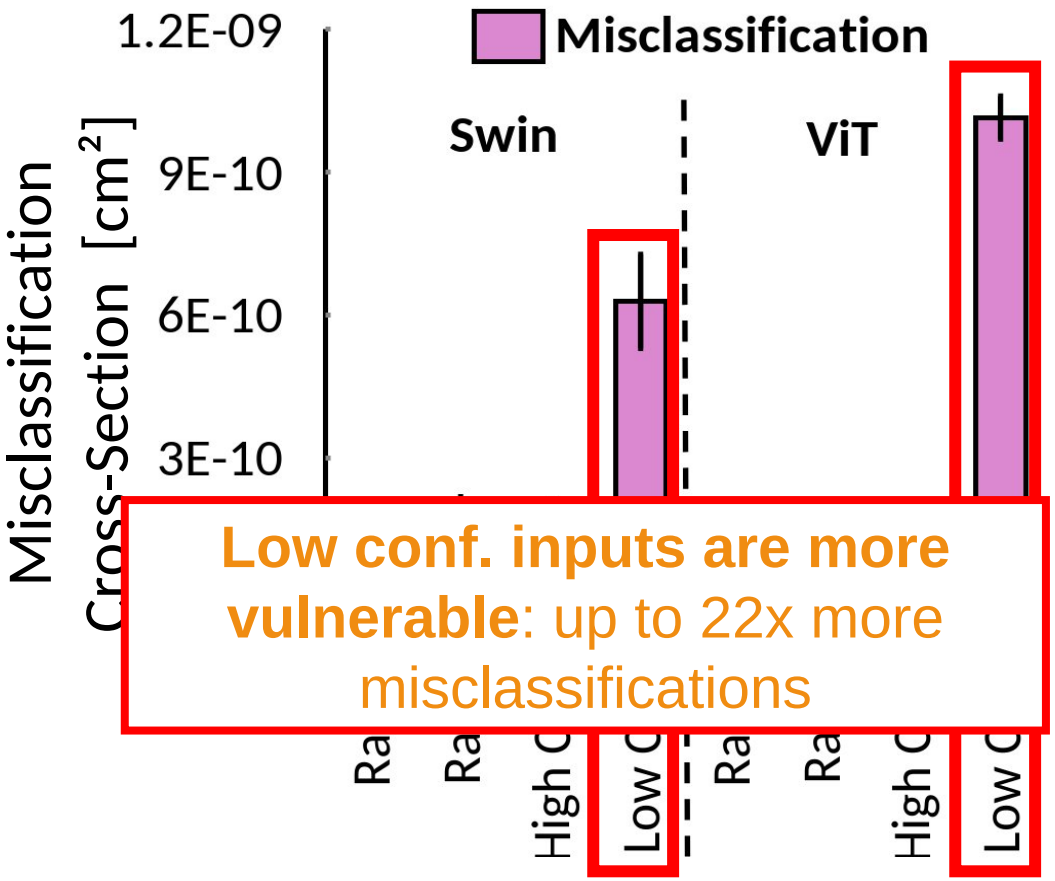
Fault simulation (protected) [2]†



[1] L. Roquet, IEEE TNS, 2026
[2] L. Roquet, IEEE LATS, 2026

Results in a nutshell

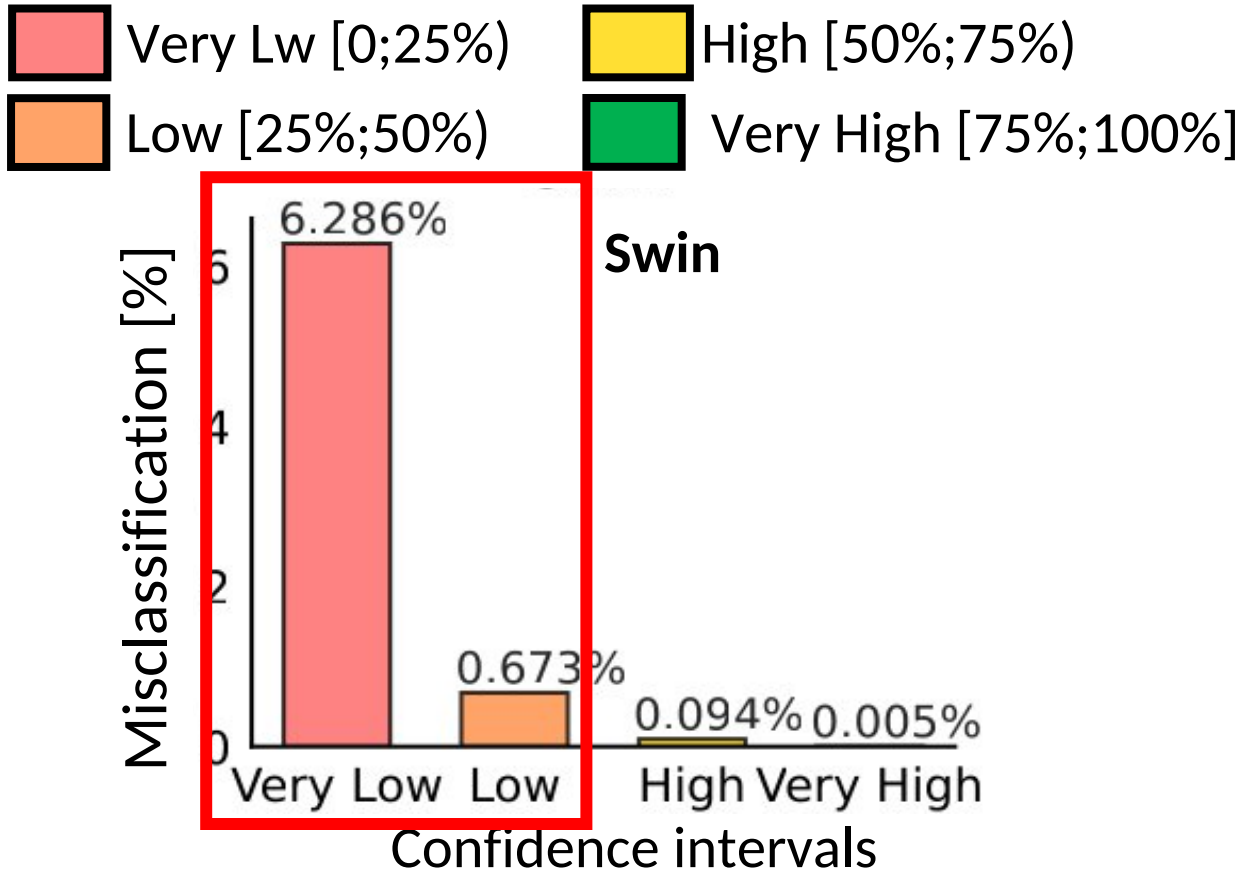
Neutron beam (unprotected) [1]*



Low conf. inputs are more vulnerable: up to 22x more misclassifications

*Same trends are observed by injecting fault at software level

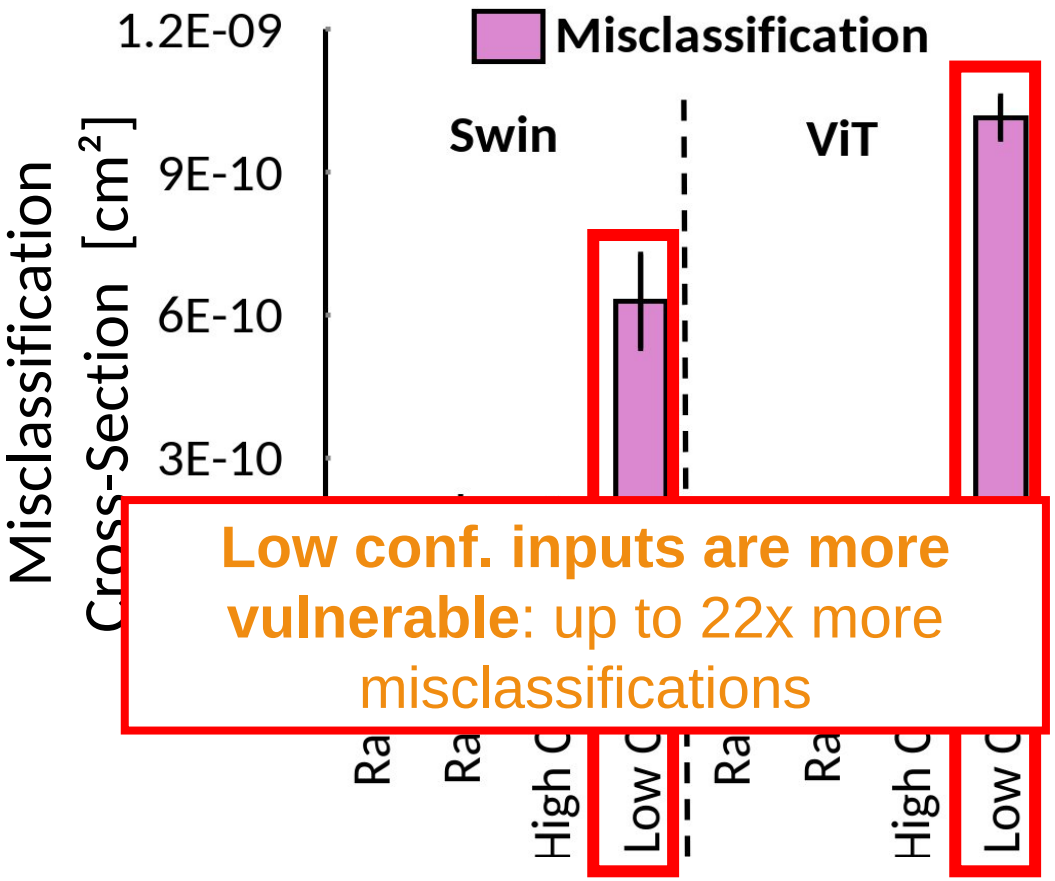
Fault simulation (protected) [2]†



[1] L. Roquet, IEEE TNS, 2026
[2] L. Roquet, IEEE LATS, 2026

Results in a nutshell

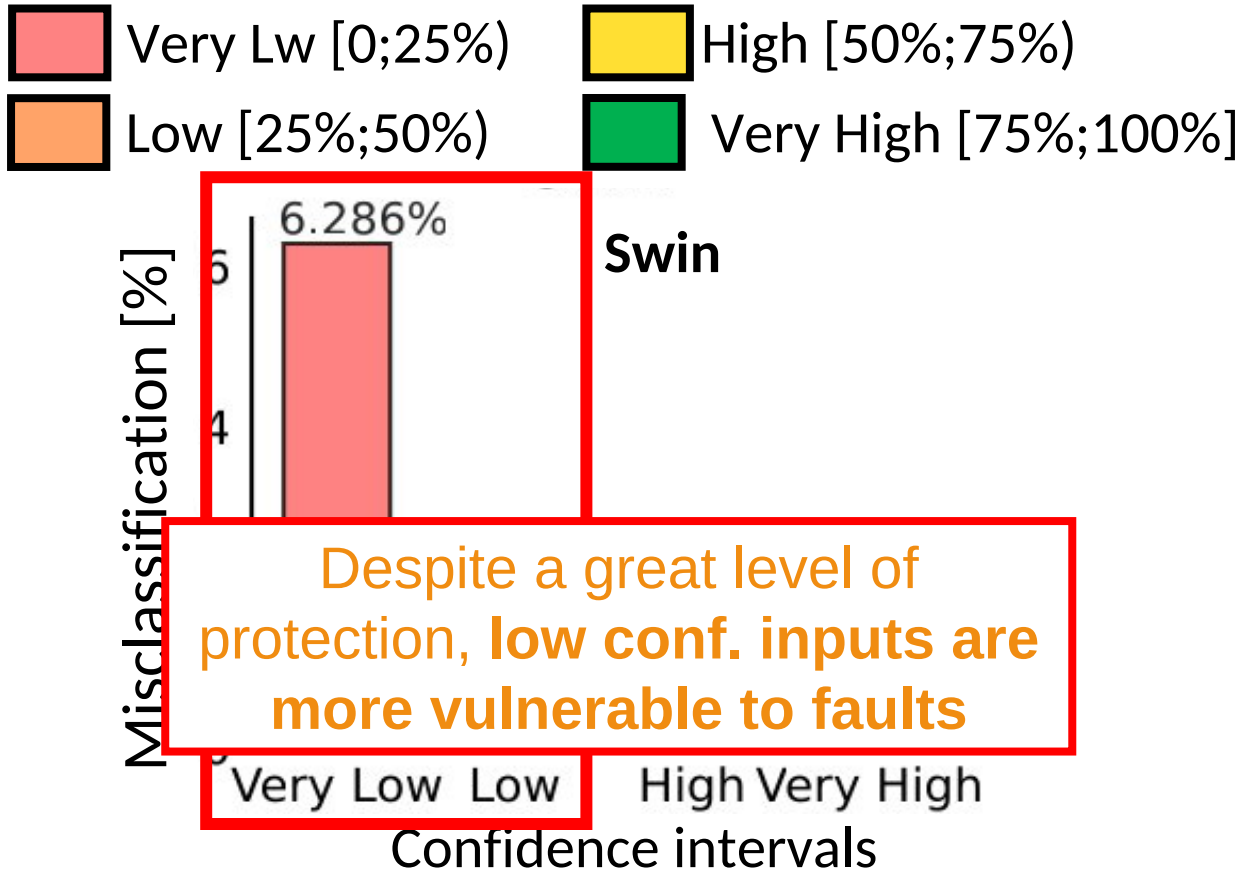
Neutron beam (unprotected) [1]*



Low conf. inputs are more vulnerable: up to 22x more misclassifications

*Same trends are observed by injecting fault at software level

Fault simulation (protected) [2]†

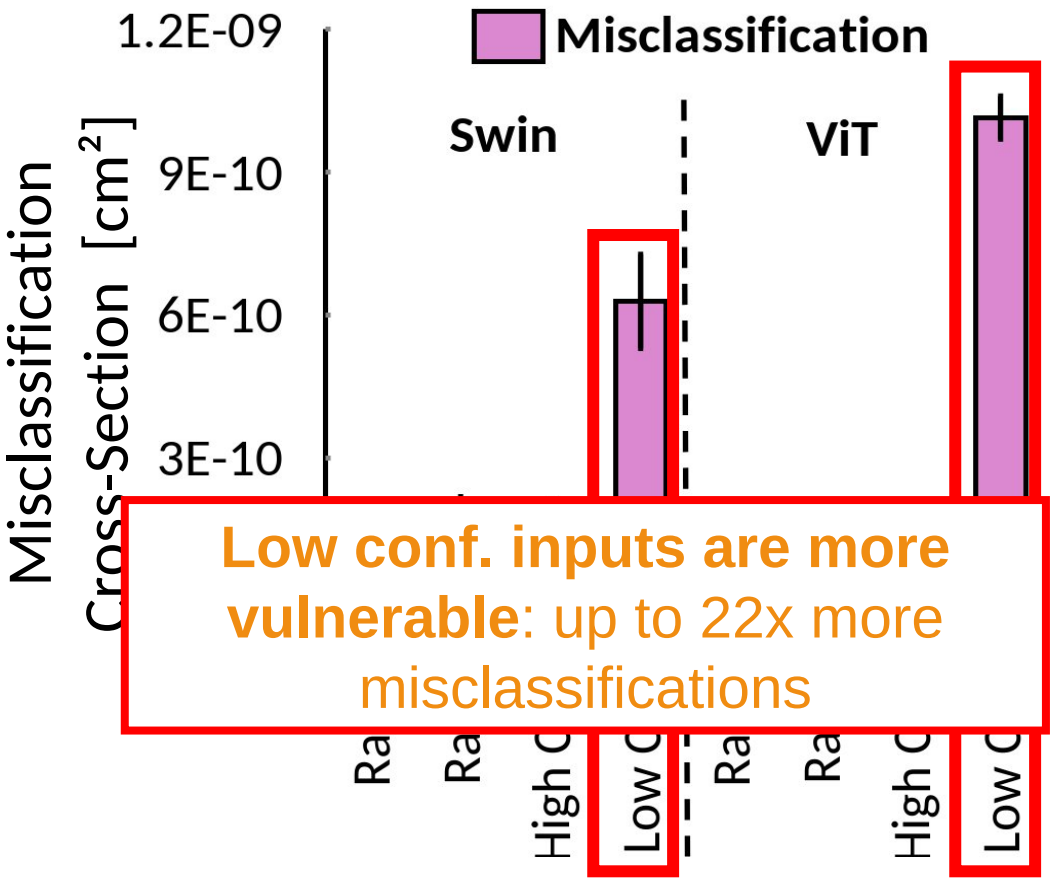


Despite a great level of protection, **low conf. inputs are more vulnerable to faults**

[1] L. Roquet, IEEE TNS, 2026
[2] L. Roquet, IEEE LATS, 2026

Results in a nutshell

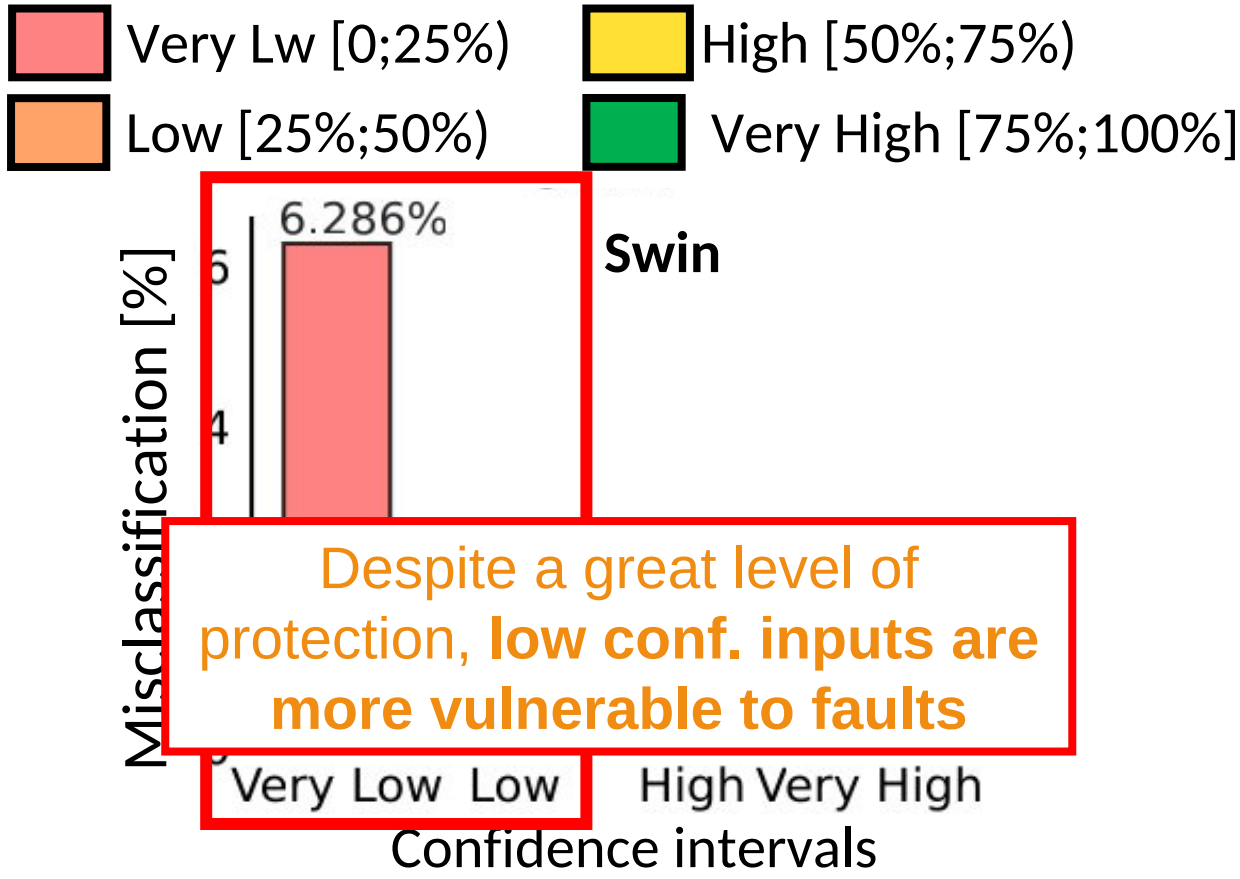
Neutron beam (unprotected) [1]*



Low conf. inputs are more vulnerable: up to 22x more misclassifications

*Same trends are observed by injecting fault at software level

Fault simulation (protected) [2]†



Despite a great level of protection, **low conf. inputs are more vulnerable to faults**

†Same trends are observed on 4 different models (CV & NLP)

[1] L. Roquet, IEEE TNS, 2026
[2] L. Roquet, IEEE LATS, 2026

Thank you

Don’t hesitate to come see my poster

CONTACT : LUCAS.ROQUET@INRIA.FR