ART: A Unified Unsupervised Framework for Incident Management in Microservice Systems

Yongqian Sun[1], Binpeng Shi[1], Mingyu Mao[1], Minghua Ma[2], Sibo Xia[1], Shenglin Zhang[1], Dan Pei[3]

[1]Nankai University, [2]Microsoft, [3]Tsinghua University

Outline

What's the incident life-cycle in microservice systems?

➔ Anomaly detection, failure triage, root cause localization

How to achieve the unification across multiple tasks?

➔ Anomalous deviation: the shared knowledge

Framework design

➔ Dependency-Aware Status Learning, Unified Failure Representation Acquisition, Unsupervised Solutions for Diagnostic Tasks

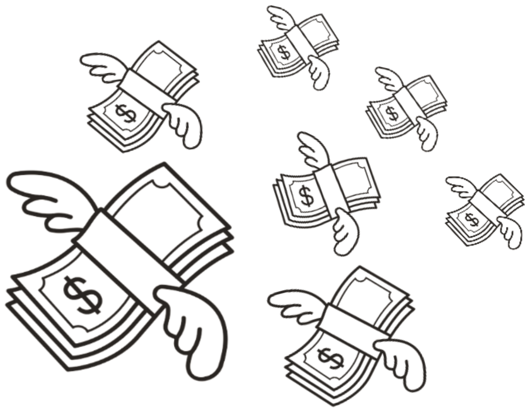
Evaluation

➔ 2 popular microservice systems

Microservice Systems

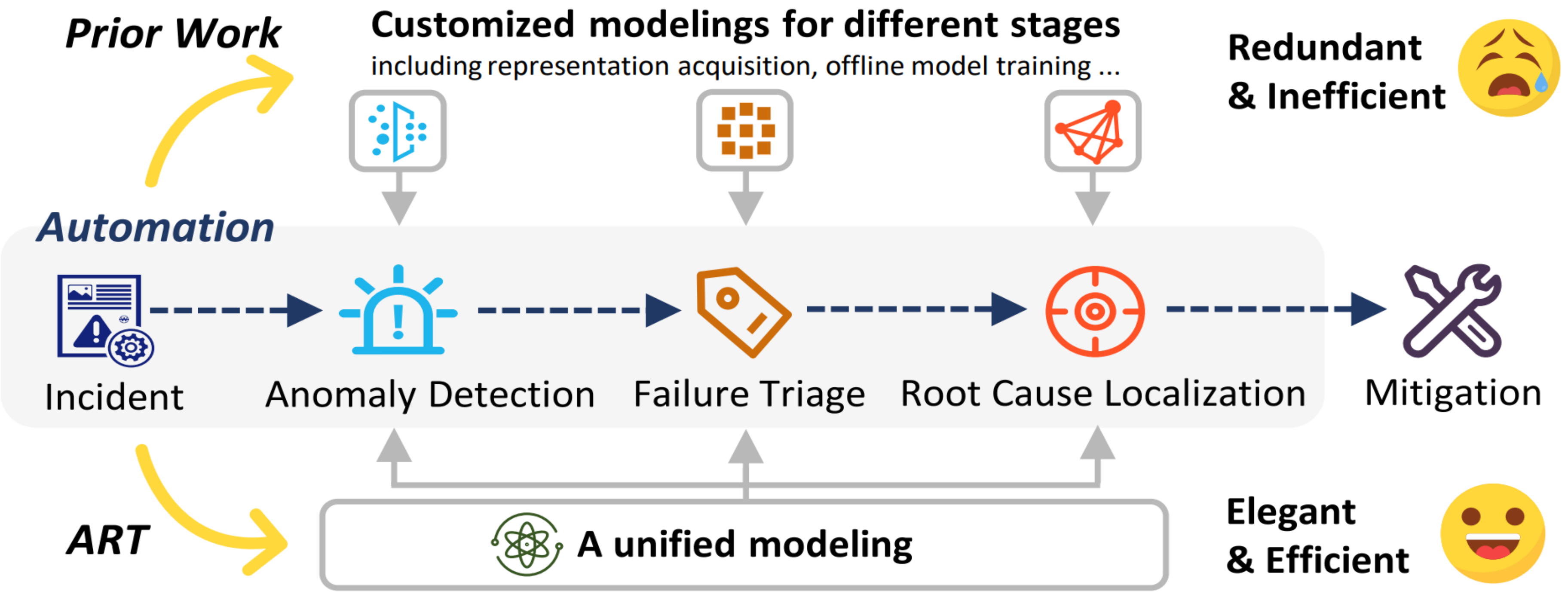
Microservice systems have become an essential part of our daily lives

Impact of Incidents



Incidents -> Unsatisfying customers -> Economic loss

An Incident Life-cycle



OCEs call for an elegant and efficient unified modeling approach

Challenges When Applying SSL

SSL

Self-supervised learning

Complexity

of the status of microservice systems

Interpretability

of representation obtained through SSL

Scarcity

of labels for downstream tasks

Outline

What's the incident life-cycle in microservice systems?

➔ Anomaly detection, failure triage, root cause localization

How to achieve the unification across multiple tasks?

➔ Anomalous deviation: the shared knowledge

Framework design

➔ Dependency-Aware Status Learning, Unified Failure Representation Acquisition, Unsupervised Solutions for Diagnostic Tasks

Evaluation

➔ 2 popular microservice systems

Design Motivation

Unification

How to achieve unification across multiple tasks?

?

What?

Is there any shared knowledge among AD, FT, and RCL?

How?

How to extract the shared knowledge effectively?

RQ1: The Shared Knowledge

- **Anomalous Deviation**
- The difference between predictions/expectations and observations

- ILD (**instance-level** deviation)
- K-dimensional vectors
- The fluctuations of each corresponding channel at instance level

Aggregation

- SLD (**system-level** deviation)
- K-dimensional vectors
- The fluctuations of each corresponding channel at system level

RQ1: The Shared Knowledge

Table 1: L_1-norms of SLDs during failure and normal hours

System Status	Metric	Deviations: $\|\|SLD\|\|_1$	Percentile
Failure Hours	Mean	100.620	P85
	Median	90.165	P73
Normal Hours	Mean	82.716	P64
	Median	77.147	P49

Deviations Manifested in Anomaly Detection

RQ1: The Shared Knowledge

Table 2: Top5 channels with the largest deviations for different failure types

Failure Type	Top5 Data Channels with the Largest Deviations				
Container Hardware	container_fs_inodes	container_fs_usage_MB	container_fs_writes	container_memory_cache	container_threads
Container Network	duration	severity_error	connection_error	service_log_other	system.net.udp.in_errors
Node CPU	system.disk.total	system.fs.inodes.free	system.fs.inodes.in_use	system.fs.inodes.total	system.load.15
Node Disk	container_last_seen	system.disk.free	system.disk.pct_usage	system.disk.total	system.disk.used
Node Memory	system.mem.pct_usage	system.mem.real.pct_useage	system.mem.real.used	system.mem.usable	system.mem.used

Deviations Manifested in Failure Triage

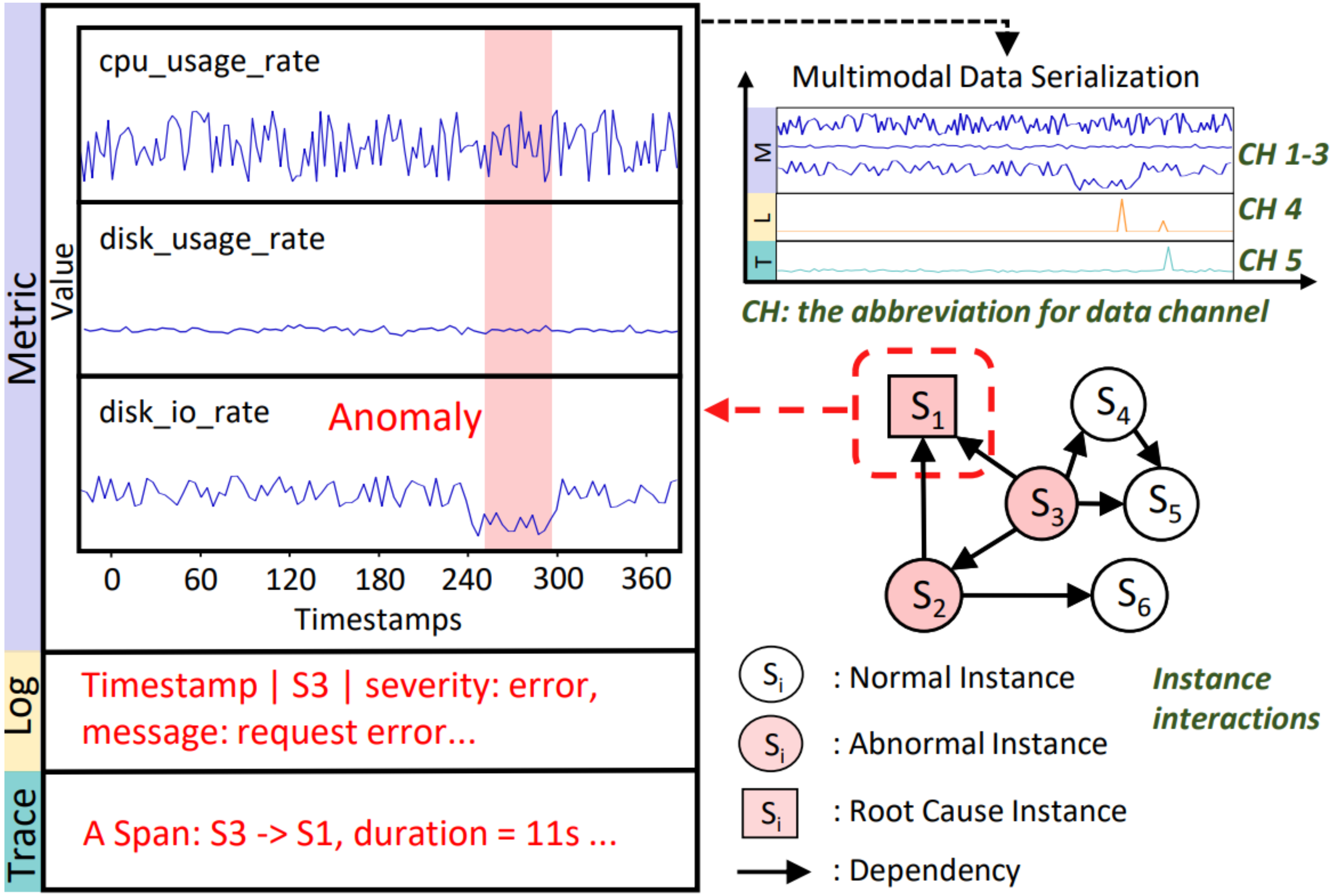
RQ1: The Shared Knowledge

Table 3: Silimarity between SLDs and ILDs of root cause and non-root cause instances

Instances	Metric	Cosine Similarity	Percentile
Root Cause	Mean	0.714	P83
	Median	0.767	P90
Non-root Cause	Mean	0.487	P46
	Median	0.499	P48

Deviations Manifested in Root Cause Localization

RQ2: How to Extract



- Channel Dependency
- Temporal Dependency
- Call Dependency

Outline

What’s the incident life-cycle in microservice systems?

➔ Anomaly detection, failure triage, root cause localization

How to achieve the unification across multiple tasks?

➔ Anomalous deviation: the shared knowledge

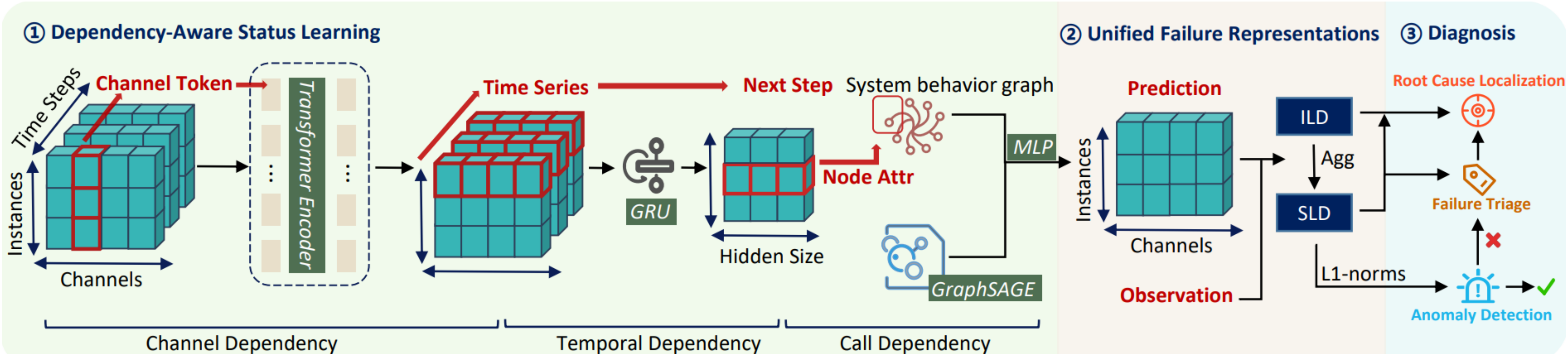
Framework design

➔ Dependency-Aware Status Learning, Unified Failure Representation Acquisition, Unsupervised Solutions for Diagnostic Tasks

Evaluation

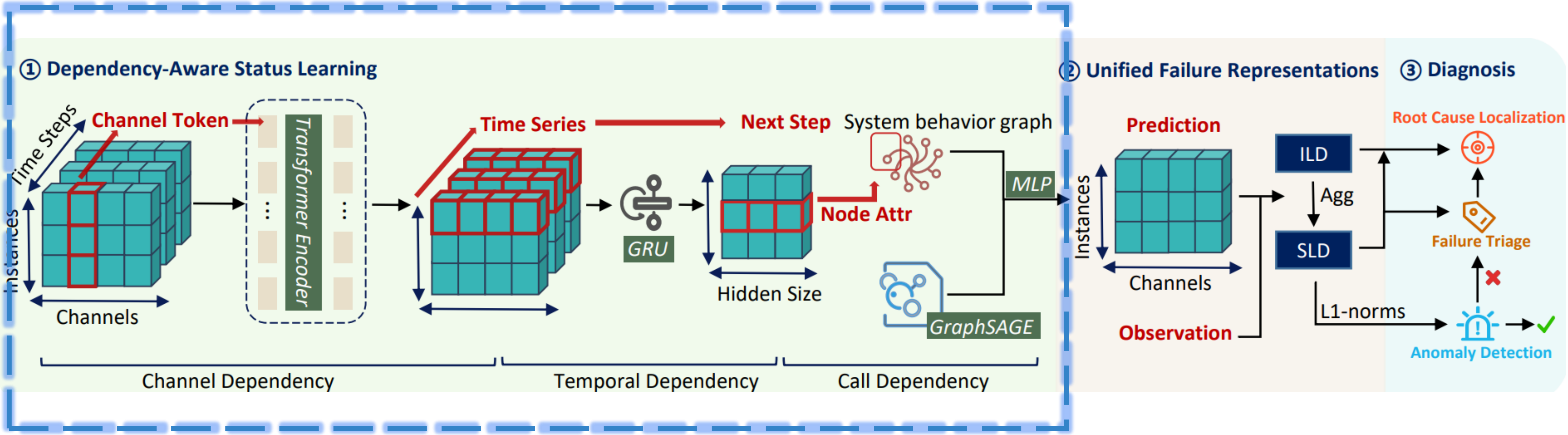
➔ 2 popular microservice systems

ART Overview



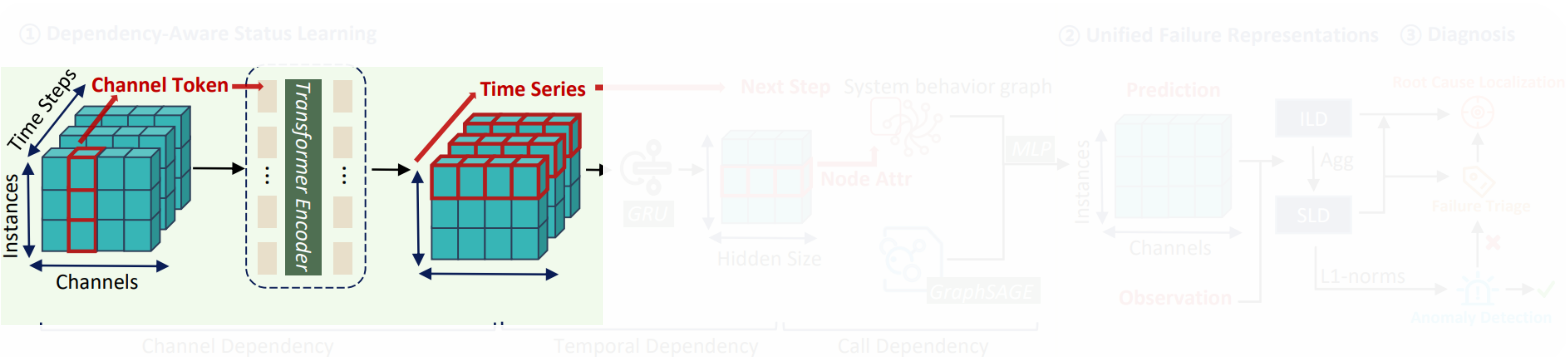
A Unified Unsupervised Framework
for Incident Management in Microservice Systems

ART - Module #1



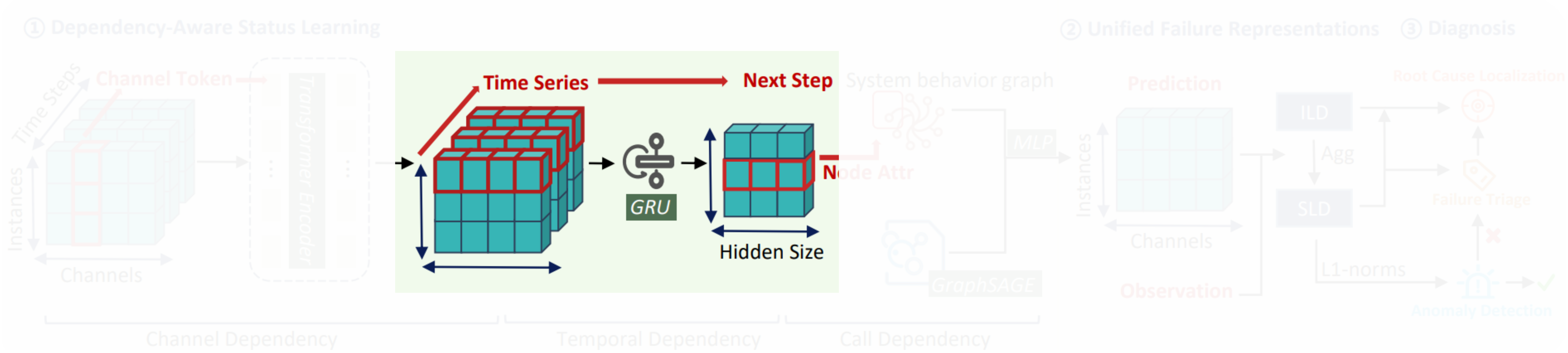
Module #1 Dependency-Aware Status Learning

ART - Module #1



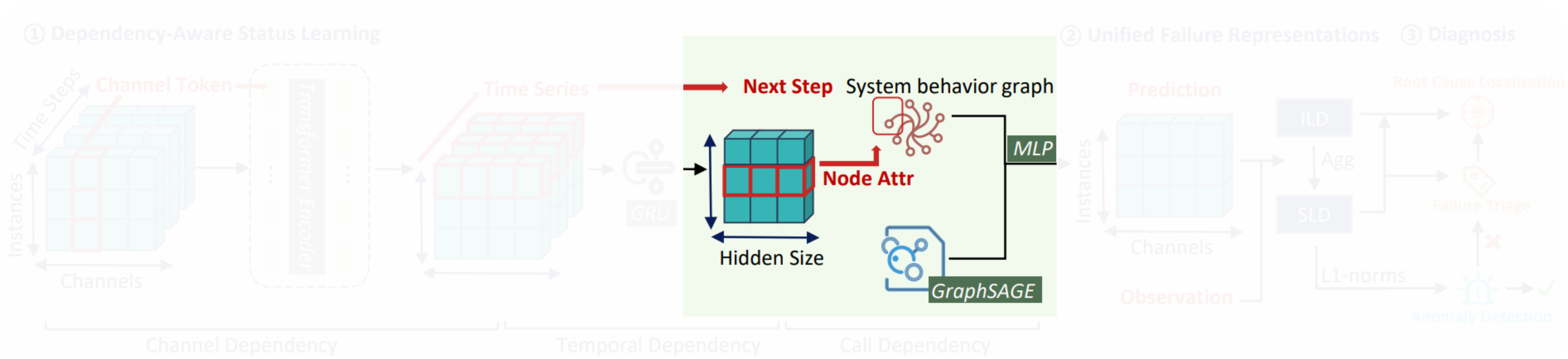
Module #1 Channel Dependency

ART - Module #1



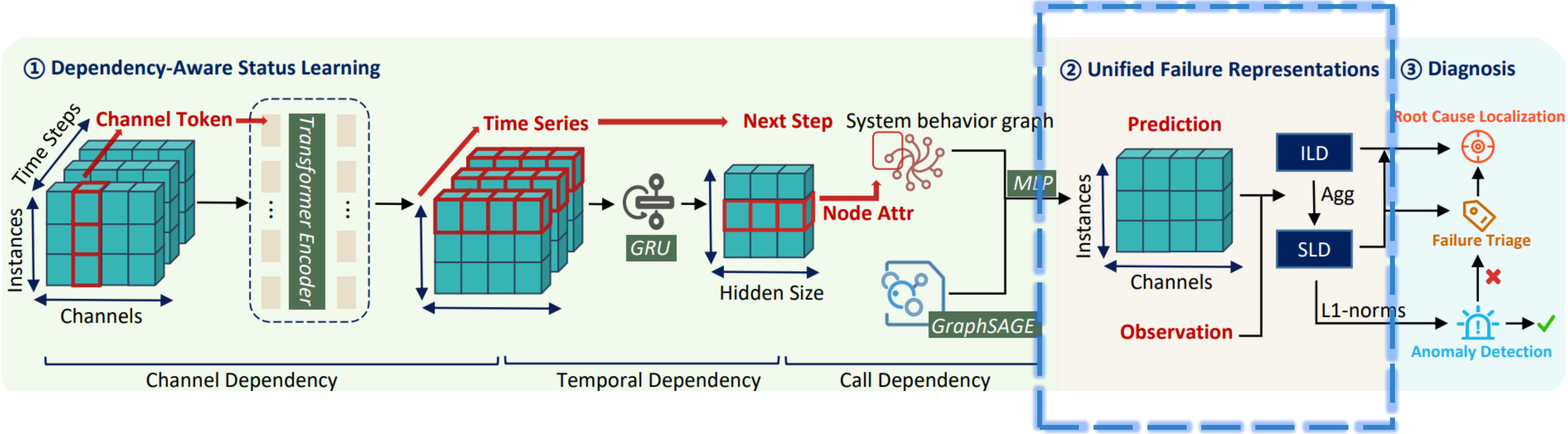
Module #1 Temporal Dependency

ART - Module #1



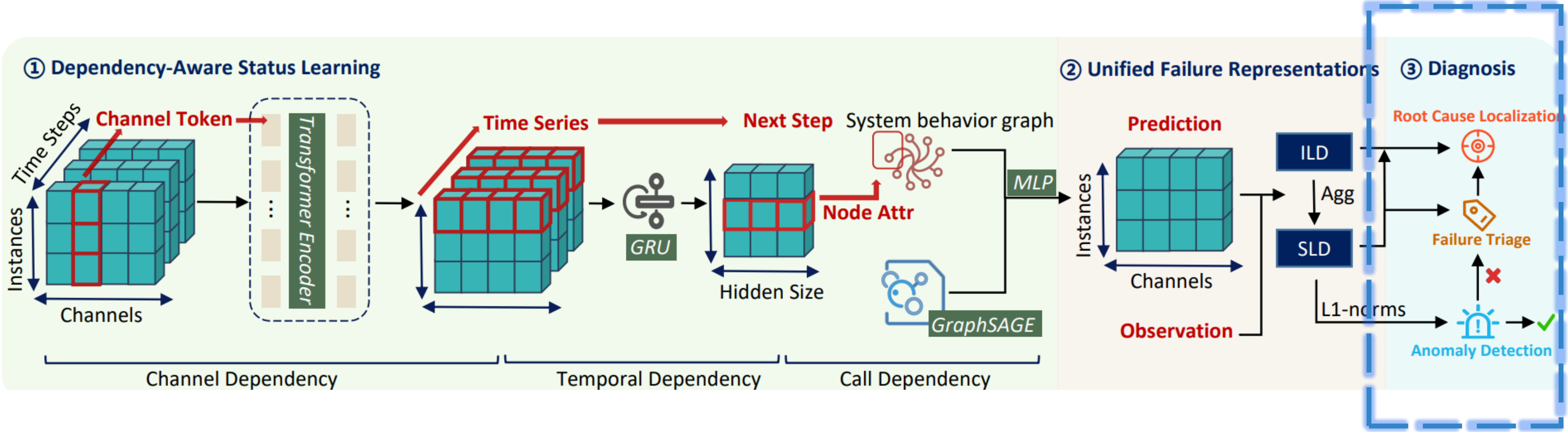
Module #1 Call Dependency

ART - Module #2



Module #2 Unified Failure Representation Acquisition

ART - Module #3



Module #3 Unsupervised Solutions for Diagnostic Tasks

Outline

What’s the incident life-cycle in microservice systems?

➔ Anomaly detection, failure triage, root cause localization

How to achieve the unification across multiple tasks?

➔ Anomalous deviation: the shared knowledge

Framework design

➔ Dependency-Aware Status Learning, Unified Failure Representation Acquisition, Unsupervised Solutions for Diagnostic Tasks

Evaluation

➔ 2 popular microservice systems

Evaluation: Performance

Table 5: Performance comparison for AD, FT, and RCL. "-" means the method does not cover the problem.

#	Method	D1									D2								
		AD			FT			RCL			AD			FT			RCL		
		Precision	Recall	F1	Precision	Recall	F1	Top1	Top3	AVG@5	Precision	Recall	F1	Precision	Recall	F1	Top1	Top3	AVG@5
multiple	ART	**0.899**	**0.990**	**0.942**	**0.836**	**0.809**	**0.812**	**0.667**	**0.810**	**0.776**	**0.877**	**0.960**	**0.917**	**0.851**	**0.796**	**0.802**	**0.722**	**0.889**	**0.870**
	Eadro [27]	0.425	0.946	0.586	-	-	-	0.137	0.315	0.302	0.767	0.935	0.842	-	-	-	0.157	0.315	0.310
	Dejevu [31]	-	-	-	0.369	0.621	0.415	0.411	0.679	0.625	-	-	-	0.718	0.340	0.417	0.402	0.667	0.619
	DiagFusion [60]	-	-	-	0.675	0.500	0.568	0.310	0.452	0.467	-	-	-	0.797	0.527	0.593	0.582	0.709	0.695
single	Hades [28]	0.866	0.863	0.865	-	-	-	-	-	-	0.867	0.868	0.868	-	-	-	-	-	-
	MicroCBR [35]	-	-	-	0.667	0.796	0.717	-	-	-	-	-	-	0.629	0.678	0.636	-	-	-
	PDiagnose [19]	-	-	-	-	-	-	0.615	0.692	0.685	-	-	-	-	-	-	0.037	0.296	0.285

Evaluation: Efficiency

Table 6: The comparison of training time (Offline) and diagnosis time (Online) per case. The unit is second. "-" means no need for training.

Method	Target			D1		D2	
	AD	FT	RCL	Offline	Online	Offline	Online
ART	✓	✓	✓	460.262	0.872	1085.767	1.363
Eadro	✓		✓	510.570	0.627	795.416	0.899
Dejavu		✓	✓	1182.468	0.427	1937.330	1.028
DiagFusion		✓	✓	621.309	4.145	310.357	3.297
Hades	✓			1214.528	0.104	2073.0413	0.415
MicroCBR		✓		-	0.278	-	0.306
PDiagnose			✓	-	4.342	-	9.919

Evaluation: Ablation Study

Table 7: The evaluation results of ablation study

Method	D1			D2		
	AD: F1	FT: F1	RCL: AVG@5	AD: F1	FT: F1	RCL: AVG@5
ART	**0.942**	**0.812**	**0.776**	**0.917**	**0.802**	**0.870**
A1	0.900	0.558	0.727	0.891	0.727	0.851
A2	0.914	0.671	0.672	0.783	0.754	0.853
A3	0.922	0.700	0.725	0.858	0.638	0.857
B1	0.936	0.794	0.748	0.906	0.717	0.855
B2	0.926	0.728	0.770	0.881	0.621	0.866
B3	0.893	0.680	0.770	0.892	0.728	0.863
B4	0.931	0.769	0.755	0.845	0.786	0.862
B5	0.893	0.758	0.714	0.888	0.570	0.844

Conclusion

- **Motivation**: OCEs call for an elegant and efficient unified modeling approach, addressing anomaly detection, failure triage, and root cause localization
- **Challenge**: complexity, interpretability, scarcity
- **Solution**: ART framework for incident management
  - Dependency-Aware Status Learning, Unified Failure Representation Acquisition, Unsupervised Solutions for Diagnostic Tasks
- **Evaluation**: superior performance with comparable efficiency

Thank you!