# Resiliency and Efficiency of Complex Enterprise Systems

Purdue University, August 24, 2023

Shiv Saini | Adobe Research

# Agenda

- Resiliency and Efficiency of Complex Enterprise Systems
  - Outage Forecasting
  - Causal Graph for microservices
  - Root cause and remediation recommendation using structured and unstructured data
- Selected Recent publications – Glimpse into team's ongoing work

# About me

- Principal Research Scientist at Adobe Research
- PhD in Econometrics, 2008, University of Wisconsin-Madison
- Current Research
  - Research topics – Causal Inference, Time Series Analysis
  - Domain – Marketing Attribution, Automated Insights, System Reliability and Efficiency
- Enterprise System Research Agenda

# Adobe Research, Bangalore

- Hire in a wide variety of fields
- Summer internship program – UG and PhD
- University Collaborations
- University Gift Funding
  - Adobe Research Gift Funding
  - Marketing Research Award

- ML for System Reliability and Efficiency
  - Generative Model Efficiency
  - NL2SQL
- User Modelling for Marketing Decisions
- Multi-modal Content Generation
- Document Understanding

# Resiliency and Efficiency of Complex Enterprise Systems

# Prevent disruptions from becoming outages

The Register

## Photostopped: Adobe Cloud evaporates in mass outage. Hope none of you are on a deadline, eh?

Photostopped: Adobe Cloud evaporates in mass outage. Hope none of you are on a deadline, eh?

3 weeks ago

YOUTUBE · **Published** December 14, 2020 9:43am EST

## Google lost $1.7M in ad revenue during YouTube outage, expert says

YouTube and other Google services, such as Gmail, suffered outage Monday morning

By **James Rogers** | **FOXBusiness** |

**Markets**

Search stocks, ETFs and more

# Performance guarantees are a promise to customers

Uptime

Latency

Data Loss

# Performance guarantees are a promise to customers

Uptime

| Availability | Downtime/Year |
| --- | --- |
| 99% | ~ 3.5 days |
| 99.9% | ~ 8 Hours |
| 99.99% | ~ 52 minutes |
| 99.999% | ~ 5 minutes |

# Microservices

- Distributed System Architecture
- Complex dependencies among components
- Scalable and flexible development
- Managed by individual self-contained teams
- Large-scale and complex architecture

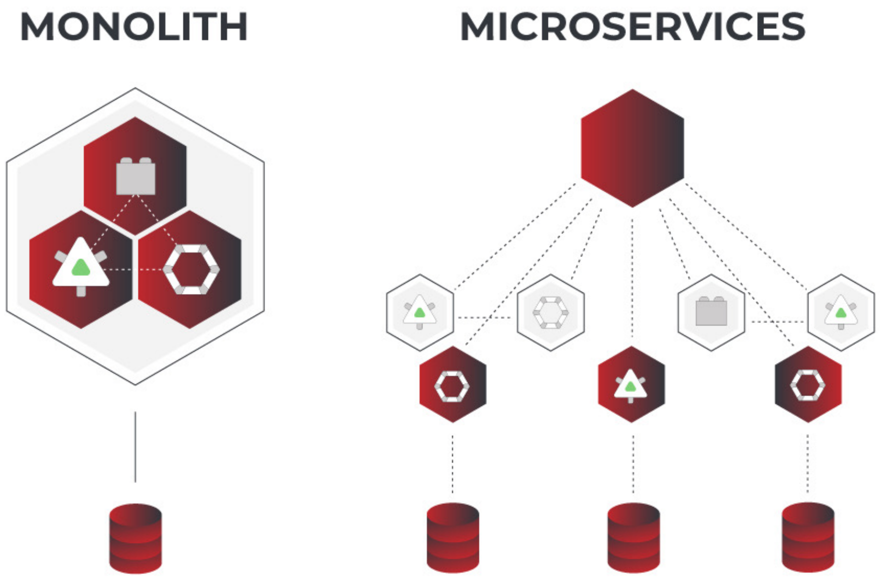



Image Source: https://codeit.us/blog/monolithic-vs-microservices-architecture

# Microservice Health Diagnosis

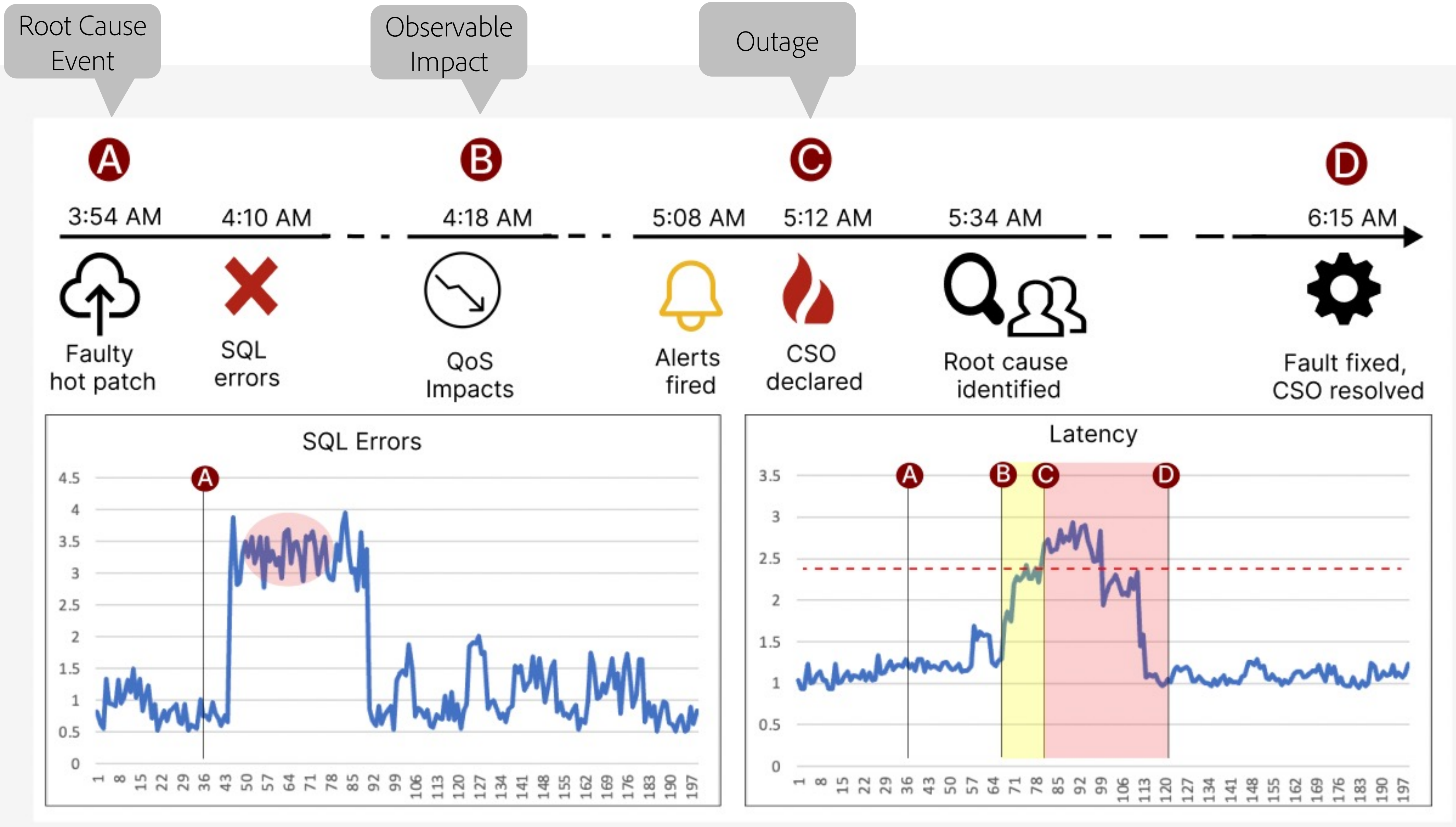
- Vulnerability is common in microservices
- Goal: Lower time to detection – Mean Time to Detection (MTTD)
- Goal: Lower time to resolution
  - Diagnosis - Structural understanding of the system
  - Resolution – How to bring the system back-up

# Can outages be forecasted?

SUDDEN OUTAGES – *NO*

OUTAGES THAT ARE NOT CAPTURED IN DATA – *NOT LIKELY*

"SLOWLY" EVOLVING ISSUES – *MAY BE*



Root Cause Event
Observable Impact
Outage
A
B
C
D
3:54 AM
4:10 AM
4:18 AM
5:08 AM
5:12 AM
5:34 AM
6:15 AM
Faulty hot patch
SQL errors
QoS Impacts
Alerts fired
CSO declared
Root cause identified
Fault fixed, CSO resolved
SQL Errors
Latency

# Outage-Watch: Early Prediction of Outages using Extreme Event Regularizer


Shubham Agarwal
Adobe Research
Bangalore, India
shagarw@adobe.com

Sarthak Chakraborty*
University of Illinois
Urbana-Champaign
Champaign, USA
sc134@illinois.edu

Shaddy Garg
Adobe
Bangalore, India
shadgarg@adobe.com

Sumit Bisht*
Amazon
Bangalore, India
bishts002@gmail.com

Chahat Jain*
Traceable.ai
Bangalore, India
chahatjain99@gmail.com

Ashritha Gonuguntla*
Cisco
Bangalore, India
ashrithag.0907@gmail.com

Shiv Kumar Saini
Adobe Research
Bangalore, India
shsaini@adobe.com

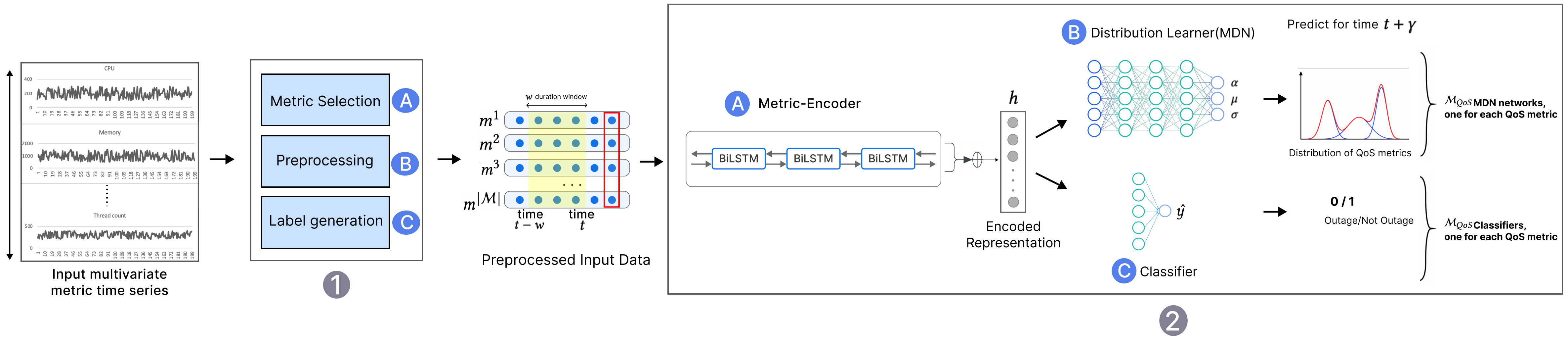

*ESEC/FSE ’23, December 3–9, 2023, San Francisco, CA, USA*

# Existing Work

- Anomaly Detection
  - Supervised and Un-supervised
- Disk failure prediction
- Outage Forecasting
  - Unsupervised – very few outages
  - Extreme event prediction
    - Extreme Value Loss - "*Modeling extreme events in time series prediction*," Ding et al, KDD 2019
  - Outages are not well-defined
  - Thresholds might not be known

# Solution

- Outage – A Quality-of-service metric (QoS) crossing user-defined threshold
  - QoS Metrics: E.g. Latency, Errors, Utilization, Queue Length
- Outage Forecasting
  - $P(QoS_t > \text{Threshold} \mid Info_{t-1})$
- Model the forecast distribution of each QoS metric as Mixture of Normals
  - Use Mixture of Density Network (MDN) Loss
- Extreme event regularizer from Ding et al, KDD 2019


CPU
Memory
Thread count
Input multivariate
metric time series
Metric Selection
A
Preprocessing
B
Label generation
C
1
w duration window
$m^1$
$m^2$
$m^3$
$m^{|\mathcal{M}|}$
time
$t-w$
time
$t$
Preprocessed Input Data
A Metric-Encoder
BiLSTM
BiLSTM
BiLSTM
$h$
Encoded
Representation
B Distribution Learner(MDN)
$\alpha$
$\mu$
$\sigma$
Predict for time $t+\gamma$
Distribution of QoS metrics
$\mathcal{M}_{QoS}$ MDN networks,
one for each QoS metric
$\hat{y}$
C Classifier
0 / 1
Outage/Not Outage
$\mathcal{M}_{QoS}$ Classifiers,
one for each QoS metric
2

# Results

| Model | Prediction Look-Ahead ($\gamma$) | | | |
|---|---|---|---|---|
| | 5 mins | 10 mins | 15 mins | 30 mins |
| Naive Bayes | 0.593 | 0.592 | 0.592 | 0.582 |
| Random Forest | 0.873 | 0.868 | 0.867 | 0.824 |
| Gradient Boost | 0.870 | 0.854 | 0.828 | 0.822 |
| BiLSTM+Classifier | 0.909 | 0.914 | 0.930 | 0.927 |
| Outage-Watch | **0.981** | **0.982** | **0.977** | **0.975** |

Synthetic Data

| Model | Precision | Recall | Reduction in MTTD | | |
|---|---|---|---|---|---|
| | | | Outage A | Outage B | Outage C |
| Naive Bayes | 1/14 | 1/3 | 24% | – | – |
| Random Forest | 1/11 | 1/3 | 0% | – | – |
| Gradient Boost | 2/12 | 2/3 | 0% | 76% | – |
| BiLSTM + Classifier | 2/10 | 2/3 | – | 56% | 26% |
| BiLSTM + MDN | 3/10 | **3/3** | 43% | 76% | 26% |
| Outage-Watch (BCE) | **3/9** | **3/3** | **54%** | 76% | **27%** |
| Outage-Watch (EVL) | **3/8** | **3/3** | 40% | **80%** | 26% |

Actual Deployment

- 2 months & 3 outages

# CausIL: Causal Graph for Instance Level Microservice Data

Sarthak Chakraborty[1], Shaddy Garg[2], Shubham Agarwal[1], Ayush Chauhan[3], Shiv Saini[1]

[1]Adobe Research India, [2]Adobe India, [3]The University of Texas at Austin

Paper Link Code Link

Proceedings of The Web Conference 2023 (WWW'23)

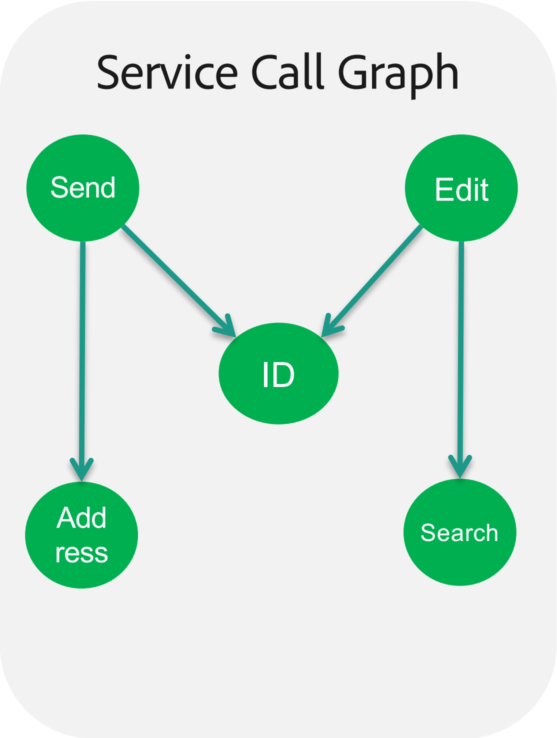
# Microservice Deployment

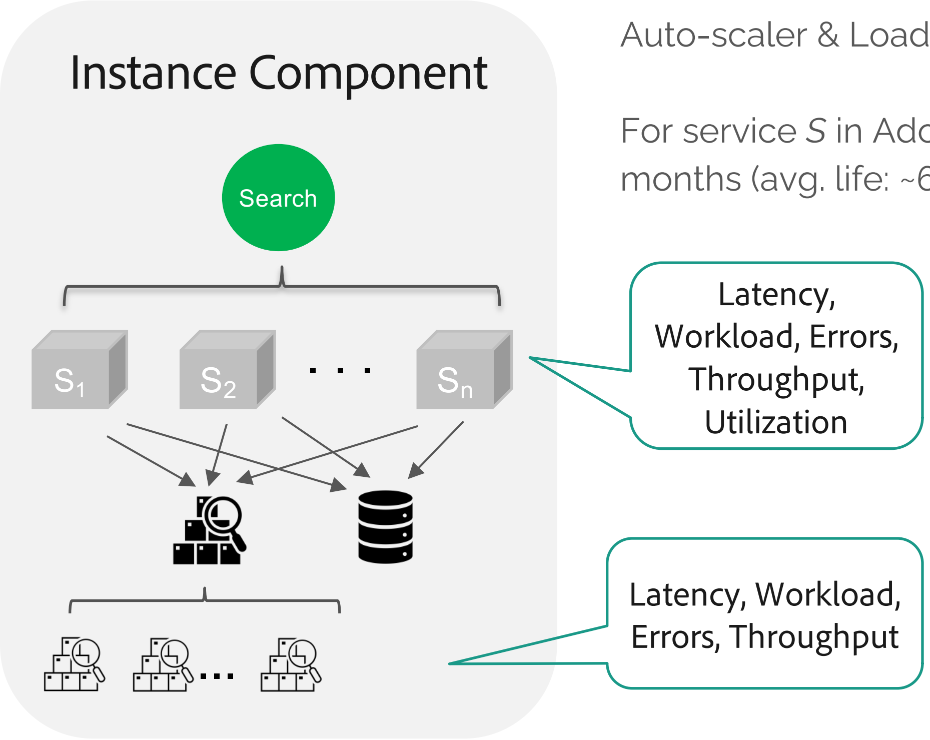
Multiple **instances**/pods of the same microservice are spawned

Unique instances are **numerous**, **transient** and **ever-changing**

Auto-scaler & Load-balancer

For service *S* in Adobe, 1117 unique instances spawned over 3 months (avg. life: ~6 hours)



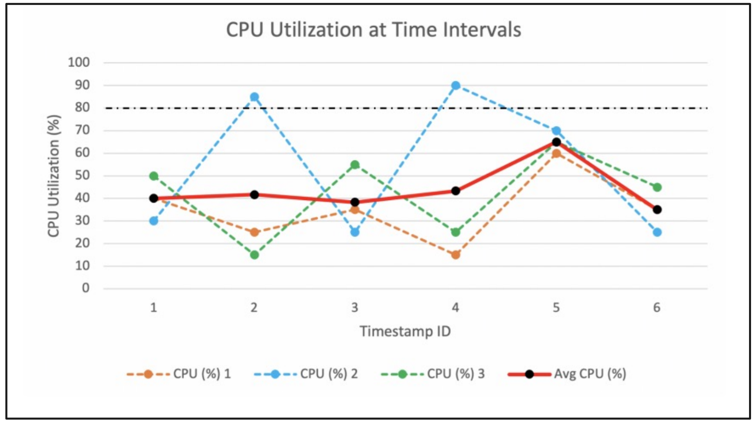



# Preliminary Approaches

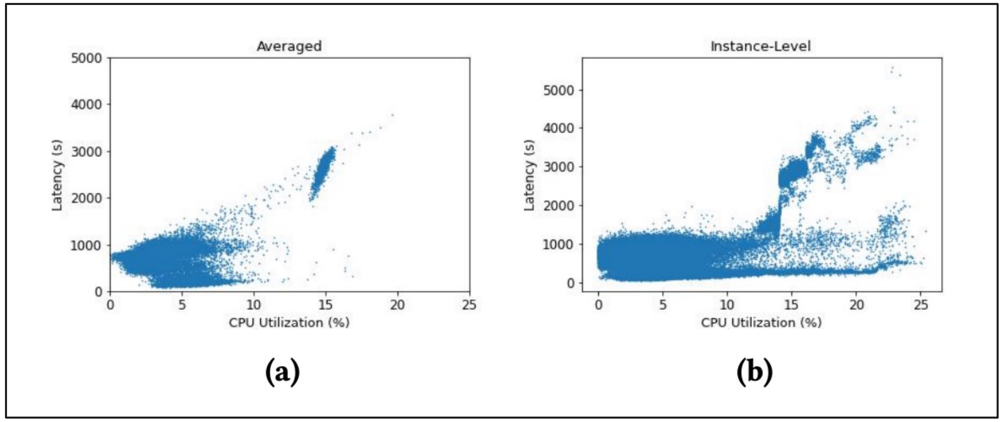
## Aggregate the metrics

Avg-fGES

- For each metric, **aggregate** the metric values across all instances for each time-step; use the aggregated value for **causal structure estimation** (fGES)




Aggregating dilutes the effect from certain instances




Changes the relationship among metrics; non-linearity not preserved

# Dependencies in Microservice

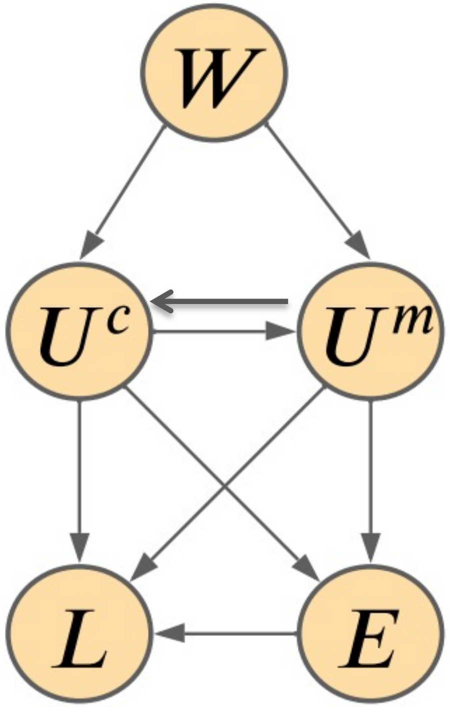
**CausIL**, a causal graph estimation methodology that considers metric variations within instances of a service:

- ✓ Can effectively model microservice deployment scenario
- ✓ Implicitly models a load-balancer and an auto-scaler
- ✓ Uses generic domain knowledge to improve efficiency
- ✓ Scalable even on addition of a new microservice

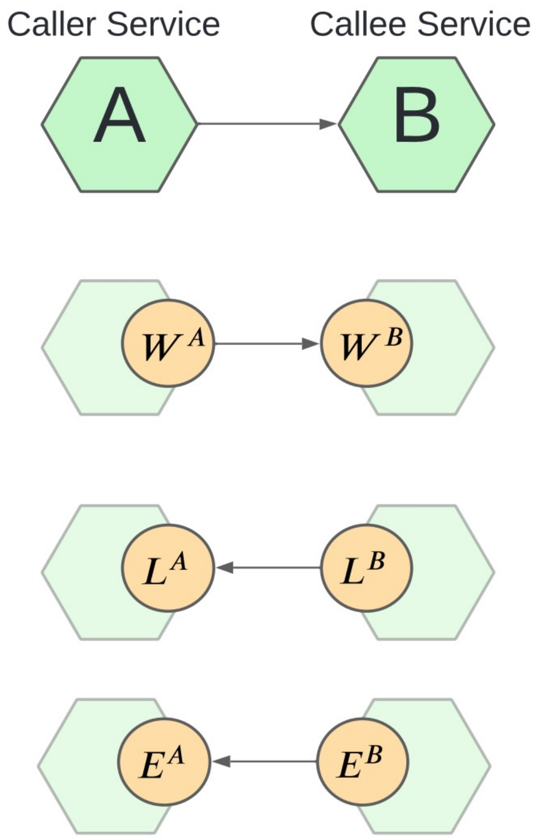
# Causal Assumptions

- Workload
- CPU Utilization
- Memory Utilization
- Latency
- Error

5 golden signals (domain expertise)

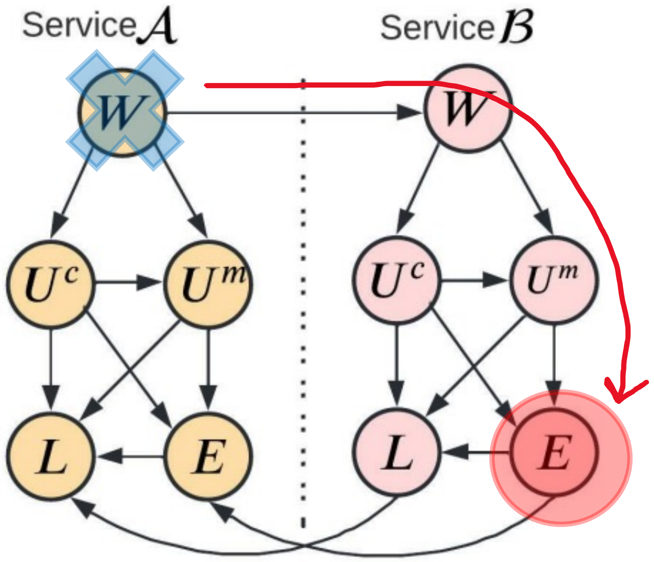



Intra-Instance

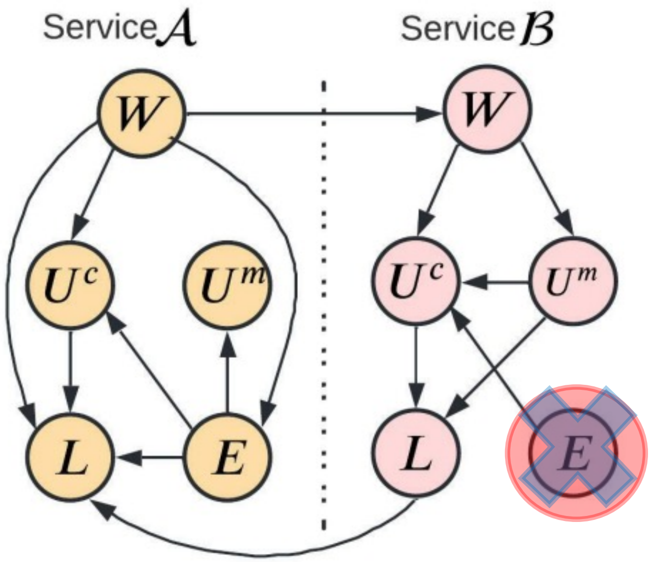



Inter-Service

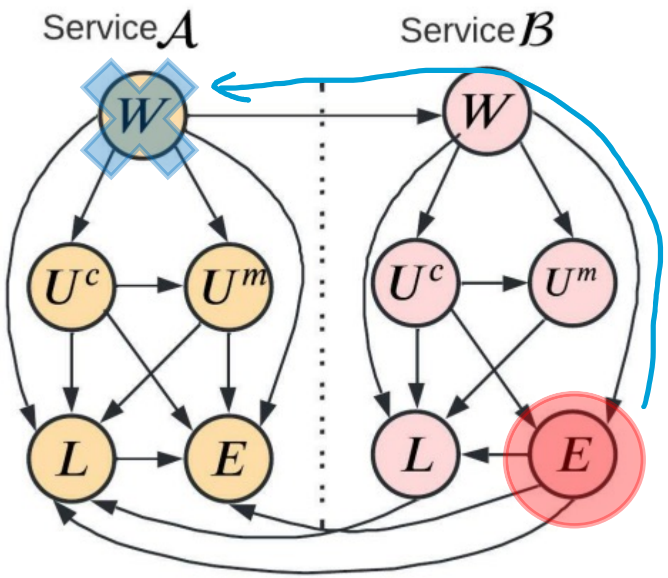
# Why new technique is needed?




Ground Truth



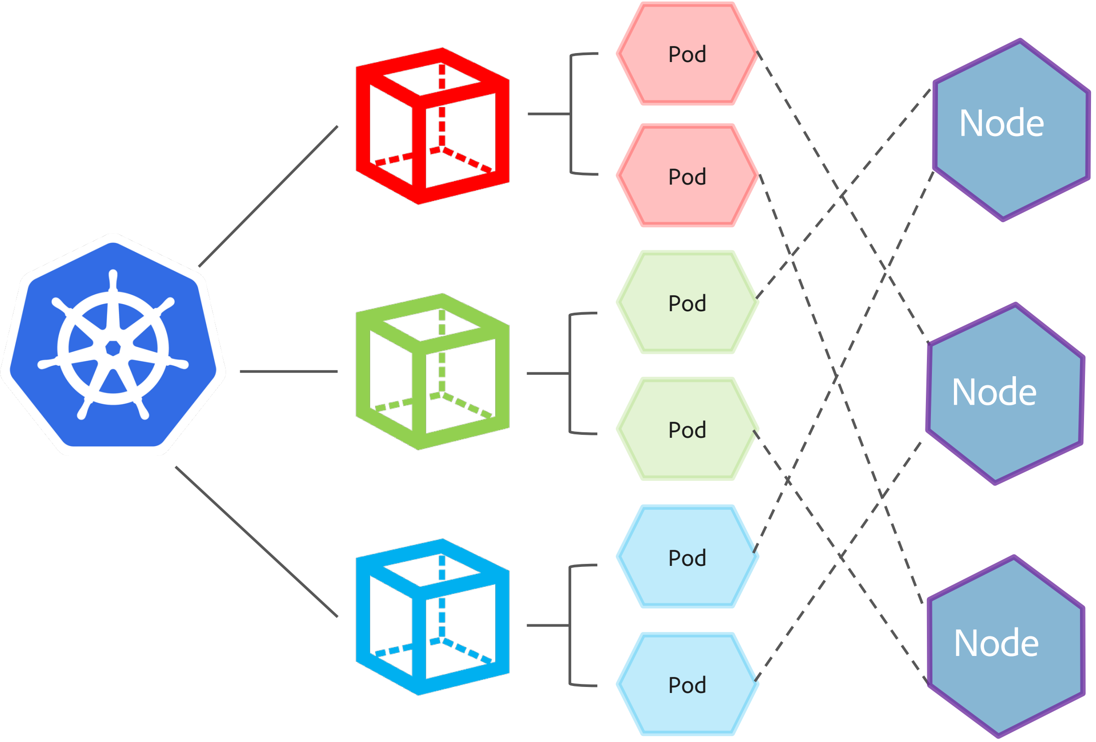

Aggregating Metric Values




Proposed Approach

# CausIL : Proposed Approach

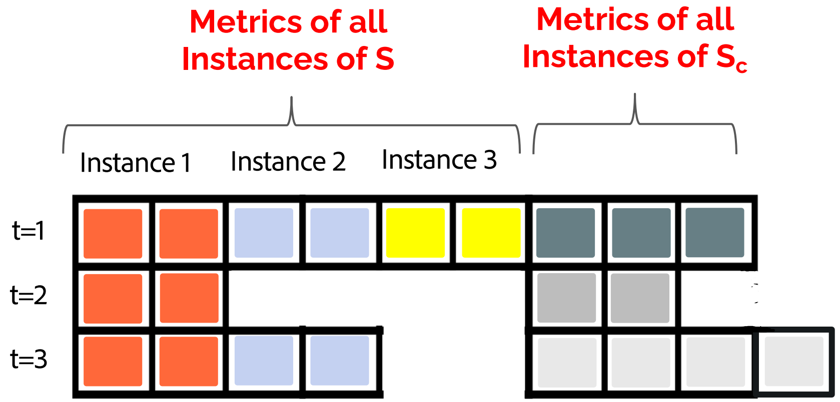
## Design Motivation

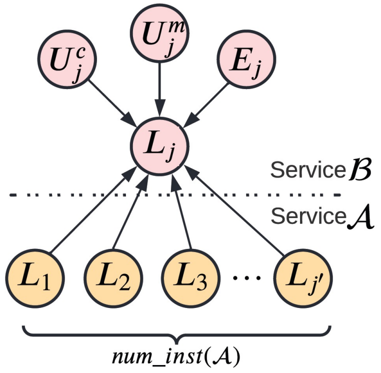



Different instances of a service are **independent** and **identical** to each other **conditioned on the load** received at the service

$$U_j | W_j \perp\!\!\!\perp U_k | W_k$$

# CausIL : Proposed Approach

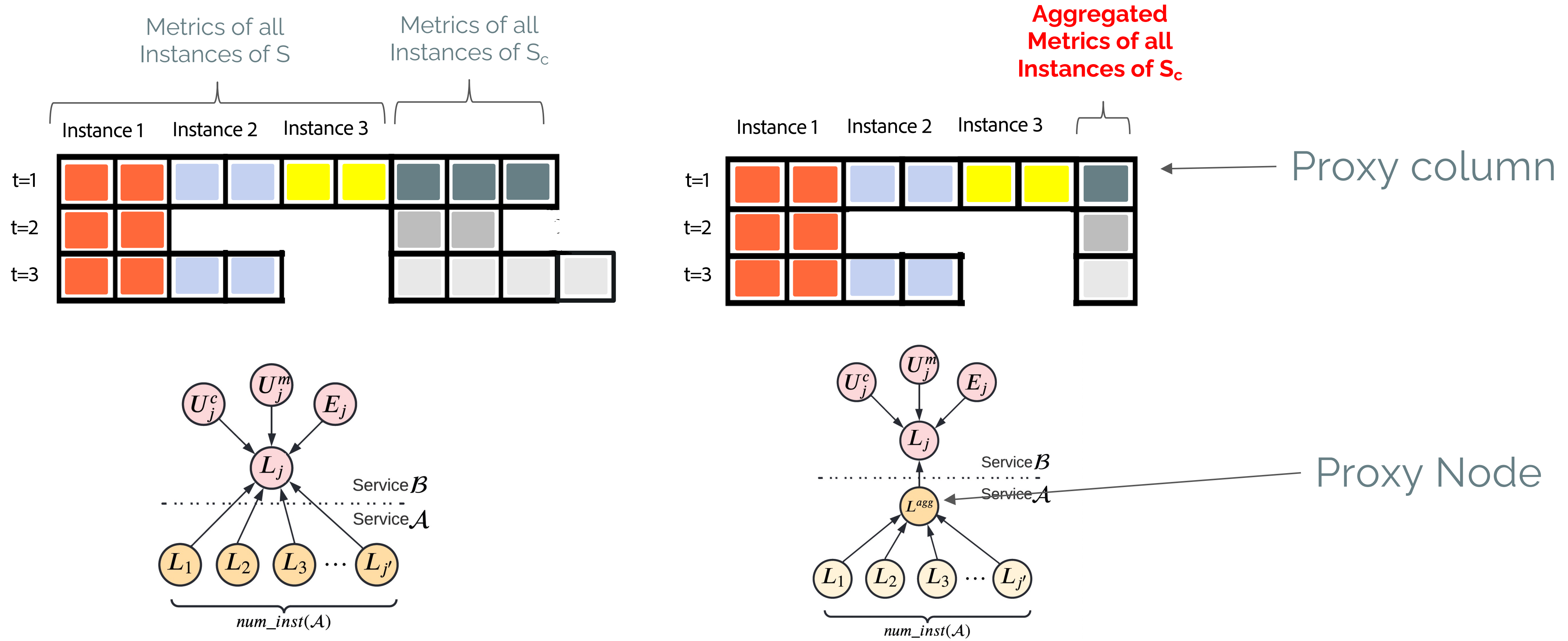


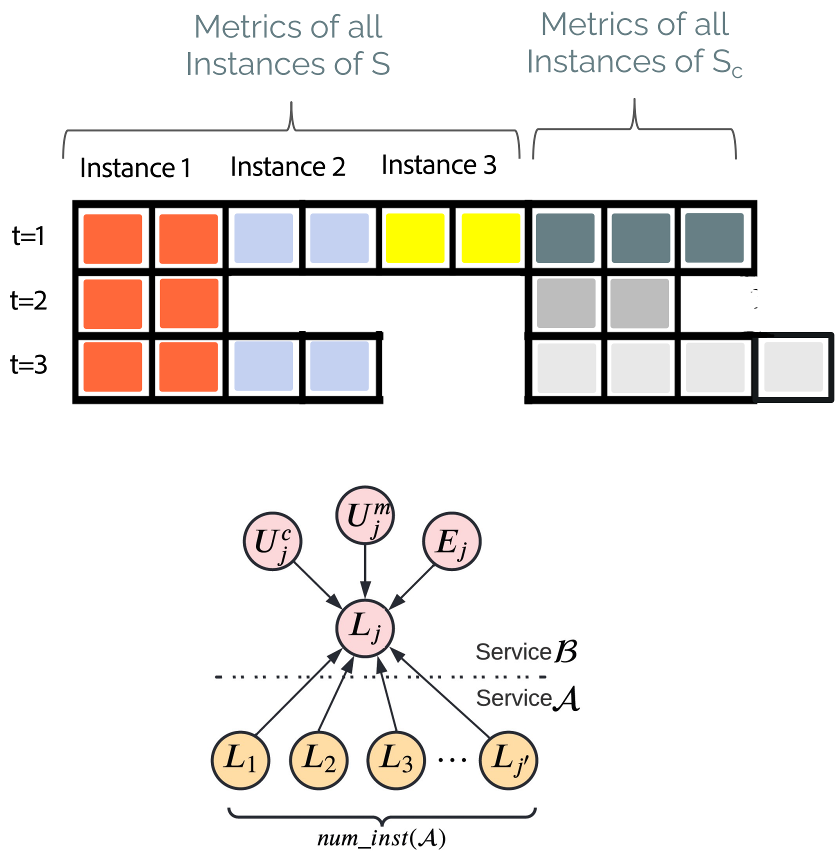

**Relevant metrics** that can form a causal structure

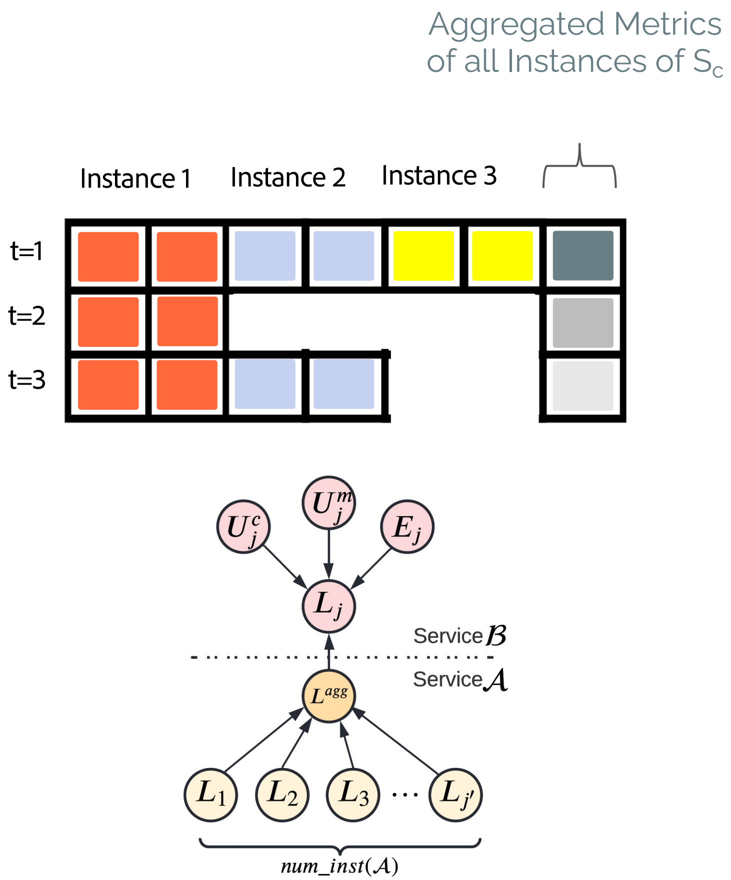
- **Golden Metrics of all instances of S**
- **Workload** for all instances of caller
- **Latency** and **Error** for all instances of callee microservices

# CausIL : Proposed Approach



# CausIL : Proposed Approach

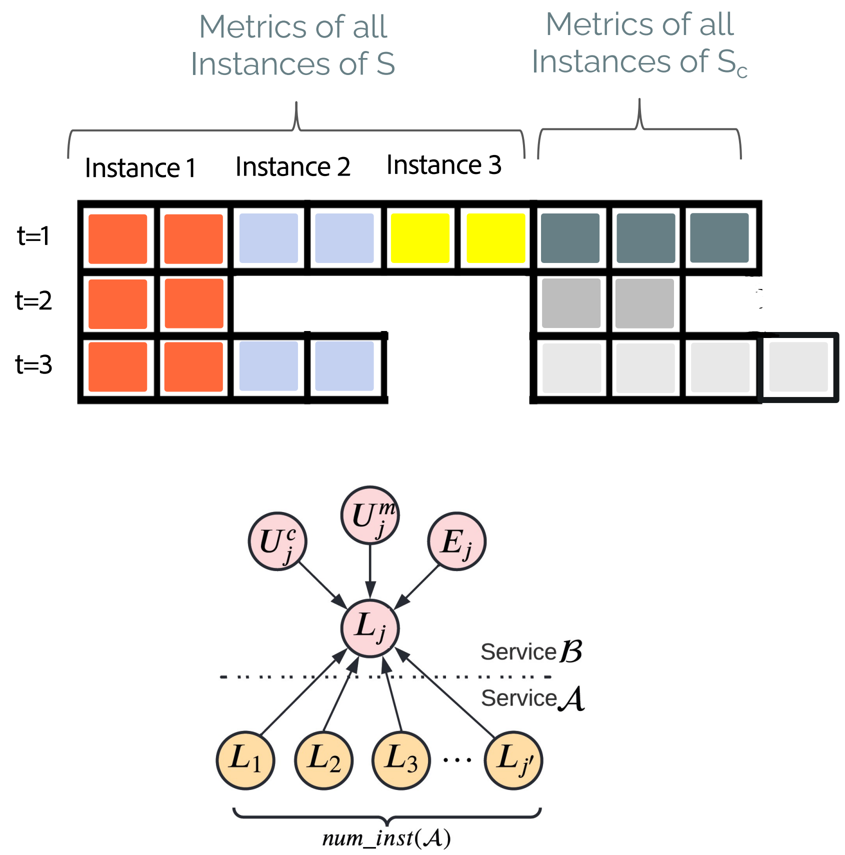


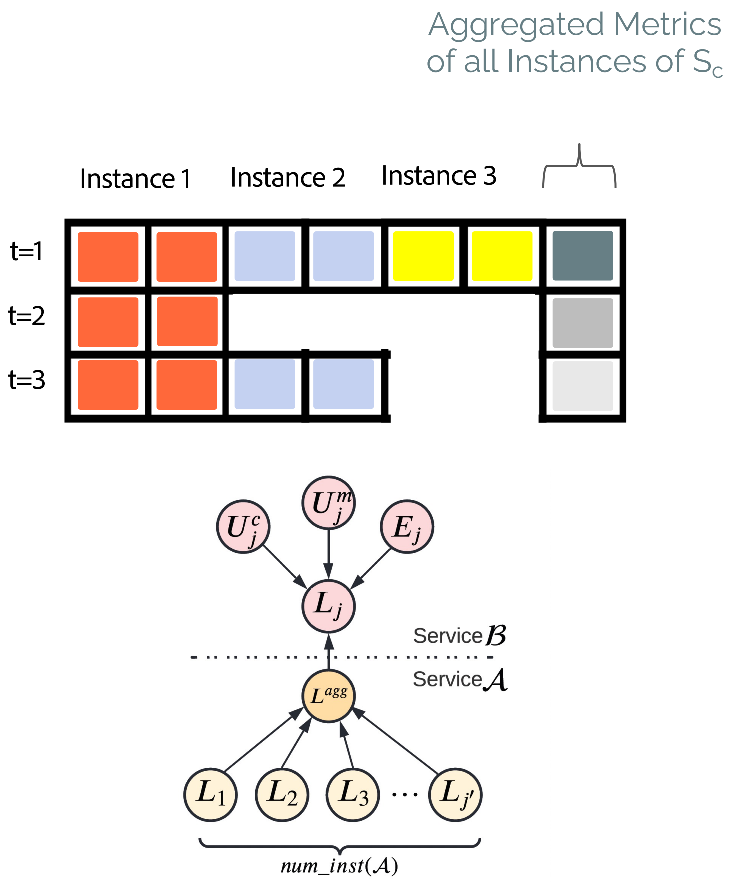



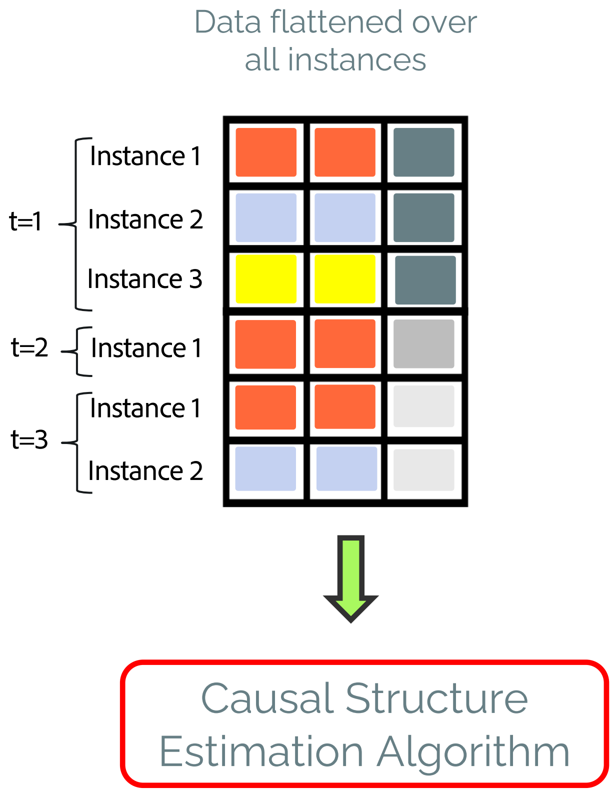


- Instances are identical and independent

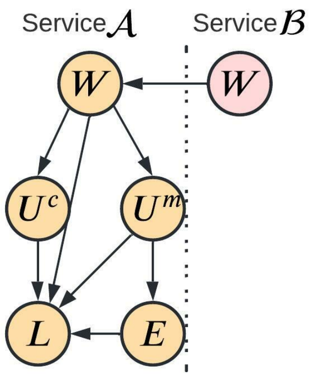
# CausIL : Proposed Approach

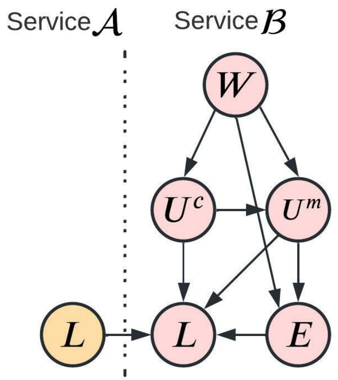


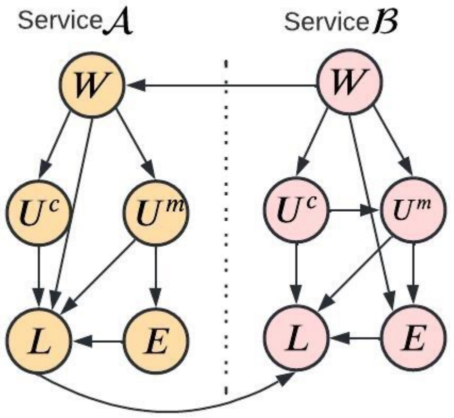



# CausIL : Proposed Approach




Causal structure estimation (fGES) of **one service at a time** and then merge

- ❑ Estimate a **function f** between the parent and the child metrics
- ❑ Choose the relationship that has **minimum BIC**; Form the edge

**BIC-based score function**

$$Score(x_{ijt}, \mathcal{P}(x_{ijt})) = -2\sum_{j,t} \log\left(L(f_i(\mathcal{P}(x_{ijt}))|x_{ijt}, \mathcal{P}(x_{ijt}))\right) + \rho k \log n_i$$

# Domain Knowledge

- **Captures metric semantics** and generic microservice architecture knowledge
- Create a **prohibited edge list**; edges that are not possible in a microservice architecture
- **Reduces time complexity** of causal estimation

| Intra-Service | Inter-Service |
| --- | --- |
| 1. No other metric within the same service affects workload<br>2. Latency does not affect resource utilization | 1. No edges across services not connected through call graph<br>2. Prohibit all edges between connected services except (a) workload in the direction of call graph, (b) latency and error in the opposite direction |

# Implementation Details

### Datasets

**Synthetic & Semi-Synthetic**

10/20/40 services

### Baselines

**Avg-fGES & CausIL**

Polynomial regression for **f** with varying degree

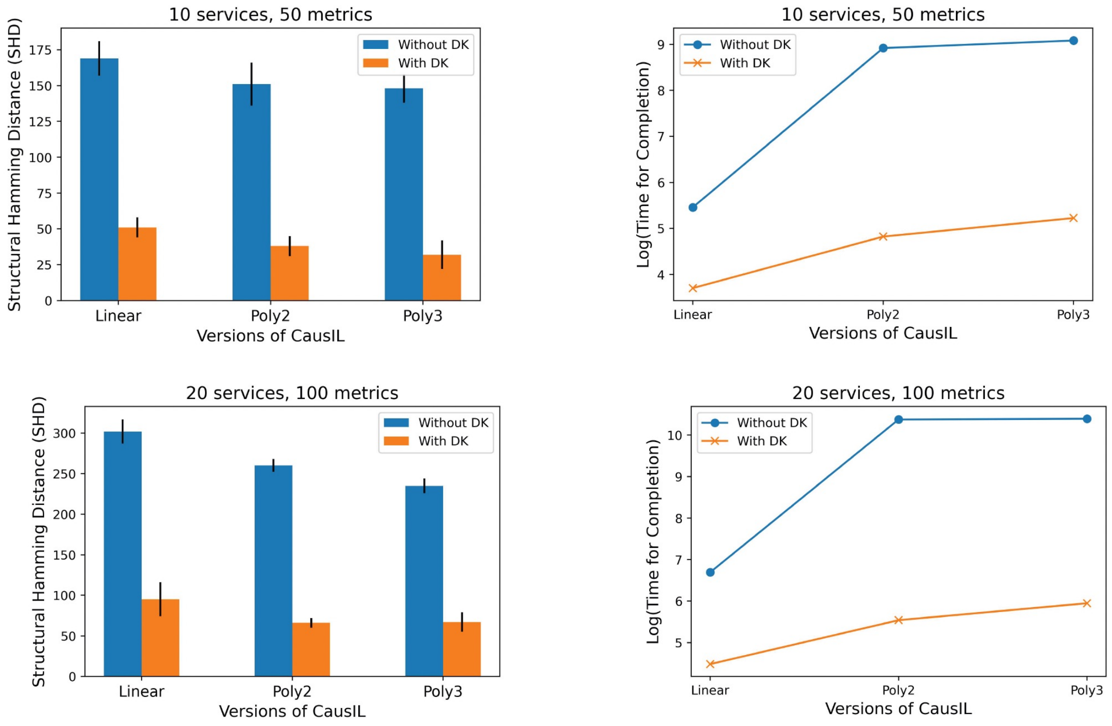
### Metrics

**Graph Comparison Metrics**

SHD, Precisions & Recall

# Evaluation Results

## Impact of Domain Knowledge




More than **3.5x** improvement in **SHD** of estimated causal graph with domain knowledge

**70x** improvement in **computation time**

Redundant edges are not considered for comparison while estimating causal graph

# Evaluation Results

## Baseline Comparison

**CausIL** performs **better** than Avg-fGES and FCI on **all the datasets**

**Polynomial Estimation Function** performs better since generated **data is polynomial**

| # Services, # Metrics | Model | $\mathcal{D}^{syn}$ SHD | AdjP | AdjR | AdjF | AHP | AHR | AHF | $\mathcal{D}^{semi-syn}$ SHD | AdjP | AdjR | AdjF | AHP | AHR | AHF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10, 50 | FCI | 53 | 0.772 | 0.841 | 0.805 | 0.704 | **0.909** | 0.793 | 53 | 0.762 | 0.873 | 0.814 | 0.69 | **0.906** | 0.783 |
| | Avg-fGES-Lin | 54 | 0.794 | 0.854 | 0.822 | 0.686 | 0.864 | 0.765 | 54 | 0.793 | 0.851 | 0.82 | 0.68 | 0.857 | 0.759 |
| | Avg-fGES-Poly2 | 48 | 0.81 | 0.861 | 0.834 | 0.723 | 0.892 | 0.799 | 50 | 0.807 | 0.845 | 0.825 | 0.713 | 0.883 | 0.789 |
| | Avg-fGES-Poly3 | 46 | 0.837 | 0.839 | 0.837 | 0.747 | 0.893 | 0.814 | 46 | 0.837 | 0.838 | 0.836 | 0.745 | 0.89 | 0.811 |
| | CausIL-Lin | 51 | 0.788 | 0.878 | 0.83 | 0.695 | 0.882 | 0.777 | 53 | 0.788 | 0.874 | 0.829 | 0.684 | 0.868 | 0.765 |
| | CausIL-Poly2 | 38 | 0.889 | 0.852 | 0.869 | 0.795 | 0.895 | 0.842 | 36 | 0.892 | **0.877** | 0.883 | 0.795 | 0.892 | **0.84** |
| | CausIL-Poly3 | **32** | **0.909** | **0.878** | **0.891** | **0.823** | 0.905 | **0.862** | **35** | **0.909** | 0.875 | **0.89** | **0.797** | 0.877 | 0.835 |
| 20, 100 | FCI | 105 | 0.773 | 0.85 | 0.81 | 0.702 | **0.909** | 0.792 | 105 | 0.772 | 0.856 | 0.811 | 0.699 | **0.906** | 0.79 |
| | Avg-fGES-Lin | 103 | 0.814 | 0.832 | 0.82 | 0.706 | 0.867 | 0.778 | 104 | 0.813 | 0.827 | 0.818 | 0.709 | 0.872 | 0.782 |
| | Avg-fGES-Poly2 | 95 | 0.823 | 0.872 | 0.847 | 0.716 | 0.87 | 0.786 | 93 | 0.826 | 0.886 | 0.855 | 0.713 | 0.864 | 0.781 |
| | Avg-fGES-Poly3 | 88 | 0.845 | 0.867 | 0.857 | 0.74 | 0.876 | 0.802 | 85 | 0.846 | 0.871 | 0.836 | 0.782 | 0.869 | 0.798 |
| | CausIL-Lin | 95 | 0.812 | 0.856 | 0.832 | 0.728 | 0.897 | 0.803 | 100 | 0.809 | 0.836 | 0.82 | 0.722 | 0.892 | 0.797 |
| | CausIL-Poly2 | **66** | 0.908 | **0.895** | **0.901** | 0.801 | 0.883 | 0.84 | **68** | 0.907 | **0.892** | **0.899** | 0.795 | 0.876 | 0.833 |
| | CausIL-Poly3 | 67 | **0.913** | 0.881 | 0.896 | **0.807** | 0.884 | **0.844** | 72 | **0.911** | 0.859 | 0.884 | **0.804** | 0.882 | **0.841** |
| 40, 200 | FCI | 204 | 0.792 | 0.814 | 0.803 | 0.719 | 0.907 | 0.803 | 206 | 0.781 | 0.802 | 0.791 | 0.706 | **0.904** | 0.793 |
| | Avg-fGES-Lin | 203 | 0.837 | 0.78 | 0.807 | 0.742 | 0.886 | 0.808 | 206 | 0.837 | 0.778 | 0.806 | 0.737 | 0.88 | 0.802 |
| | Avg-fGES-Poly2 | 197 | 0.853 | 0.782 | 0.815 | 0.749 | 0.879 | 0.809 | 200 | 0.852 | 0.779 | 0.813 | 0.746 | 0.876 | 0.806 |
| | Avg-fGES-Poly3 | 204 | 0.862 | 0.741 | 0.795 | 0.763 | 0.886 | 0.82 | 203 | 0.862 | 0.746 | 0.799 | 0.759 | 0.882 | 0.816 |
| | CausIL-Lin | 186 | 0.835 | 0.811 | 0.824 | 0.759 | 0.897 | 0.827 | 188 | 0.833 | 0.811 | 0.822 | 0.755 | 0.905 | 0.823 |
| | CausIL-Poly2 | **152** | 0.919 | **0.828** | **0.871** | 0.825 | 0.899 | 0.86 | **155** | 0.92 | **0.817** | **0.864** | 0.809 | 0.879 | 0.843 |
| | CausIL-Poly3 | 160 | **0.922** | 0.783 | 0.846 | **0.83** | **0.909** | **0.863** | 162 | **0.922** | 0.784 | 0.847 | **0.821** | 0.89 | **0.854** |

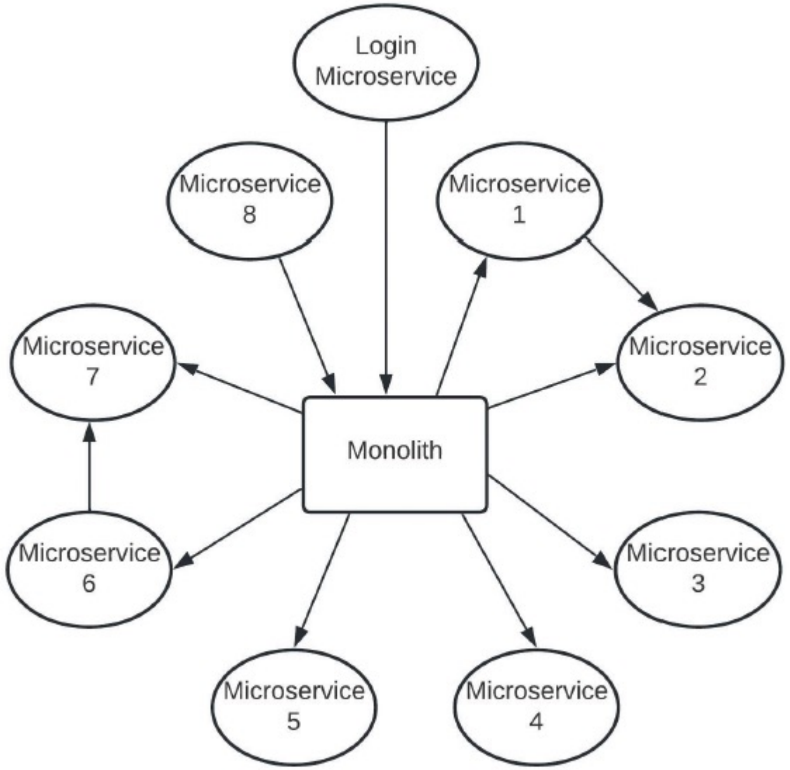
**Table 1: Comparison of CausIL against the baselines for all types of datasets.**

# Evaluation Results

## Real Data

- ➢ Data collected for a span of 2 months with 5 min granularity
- ➢ Ground Truth Causal Graph based on the causal assumptions

| Model | SHD | AdjP | AdjR | AdjF | AHP | AHR | AHF |
|---|---|---|---|---|---|---|---|
| FCI | 59 | 0.756 | 0.796 | 0.775 | 0.697 | 0.922 | 0.794 |
| Avg-fGES-Lin | 52 | 0.829 | 0.858 | 0.843 | 0.692 | 0.835 | 0.757 |
| Avg-fGES-Poly2 | 53 | 0.823 | 0.823 | 0.823 | 0.708 | 0.86 | 0.777 |
| Avg-fGES-Poly3 | 51 | **0.852** | 0.814 | 0.833 | 0.722 | 0.848 | 0.78 |
| CausIL-Lin | 50 | 0.807 | 0.814 | 0.81 | 0.737 | 0.913 | 0.816 |
| CausIL-Poly2 | **40** | 0.818 | **0.876** | **0.846** | **0.785** | **0.96** | **0.864** |
| CausIL-Poly3 | 46 | 0.824 | 0.867 | 0.845 | 0.739 | 0.898 | 0.811 |



# Conclusion

- Causal Structure Estimation for microservices when multiple instances of a microservice are deployed
  - Instances are dynamic and transient in nature
- Domain Knowledge improves the performance as well as computation complexity
  - Domain Knowledge generic to any microservice architecture
  - Written as general rules
- Estimates causal relationship one service at a time
  - Scales linearly on adding a new service
  - Each service takes 12-13s with standard deviation of 1.09s on average

# *ESRO*: Experience Assisted Service Reliability against Outages

Sarthak Chakraborty[†*], Shubham Agarwal[‡], Shaddy Garg[§],
Abhimanyu Sethia[¶*], Udit Narayan Pandey[¶*], Videh Aggarwal[¶*], Shiv Saini[‡]

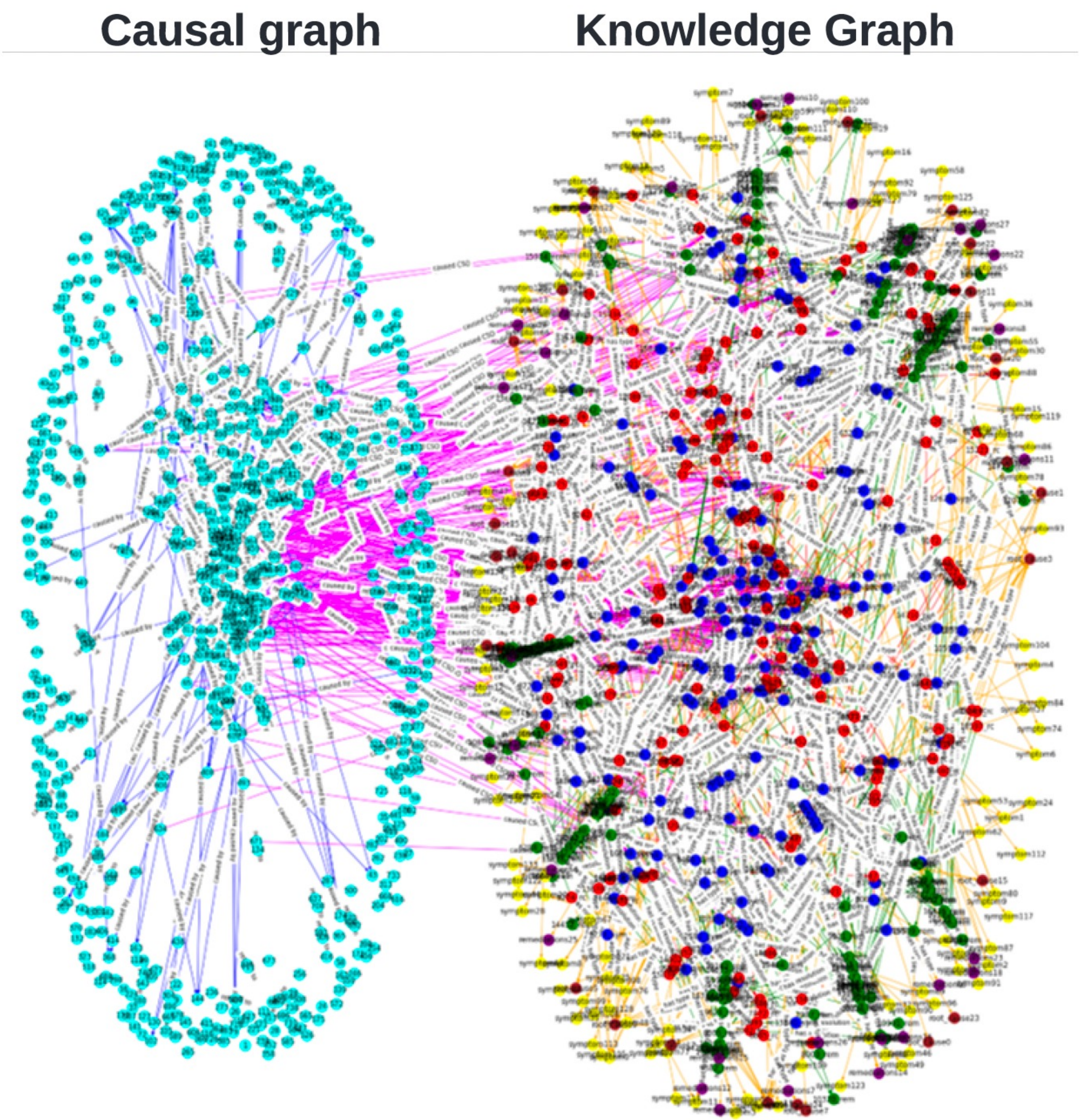
[†]*University of Illinois Urbana-Champaign, USA*, [‡]*Adobe Research, India*,
[§]*Adobe, India*, [¶]*Indian Institute of Technology Kanpur, India*

sc134@illinois.edu, shagarw@adobe.com, shadgarg@adobe.com, abhimanyusethia12@gmail.com,
udit.pusp@gmail.com, videh1aggarwal@gmail.com, shsaini@adobe.com

*ASE ’23, September 11–15, 2023, Luxembourg*

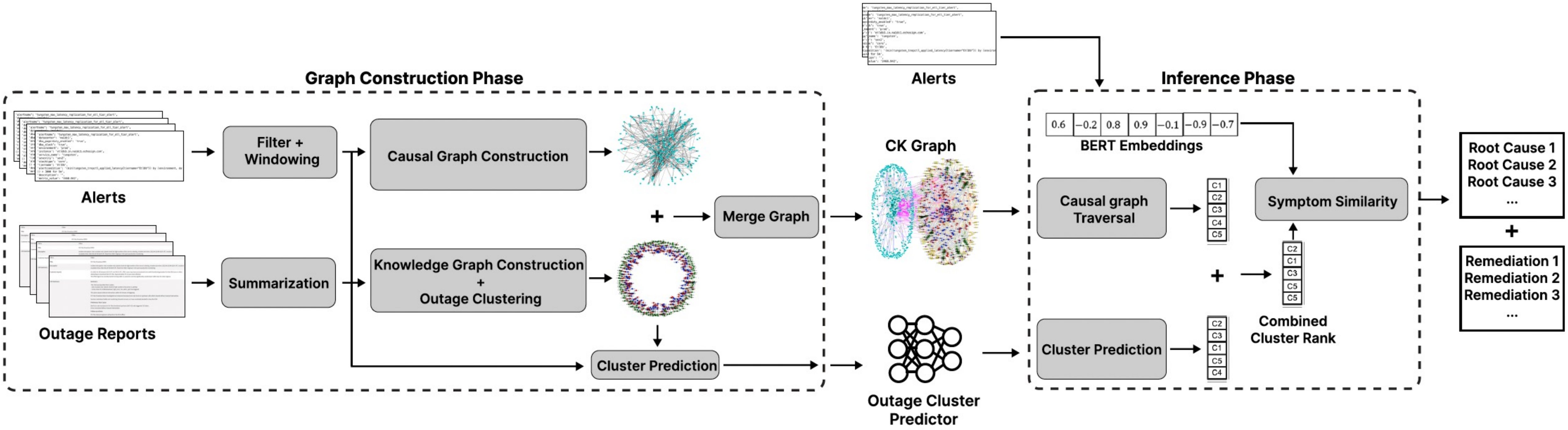
Metric Alerts

- *Real-time*
- *Difficult to interpret*




Incident Reports

- *Stale*, but
- *Semantically rich*


Graph Construction Phase
Alerts
Filter + Windowing
Causal Graph Construction
Outage Reports
Summarization
Knowledge Graph Construction + Outage Clustering
Merge Graph
Cluster Prediction
Alerts
CK Graph
Outage Cluster Predictor
Inference Phase
0.6 −0.2 0.8 0.9 −0.1 −0.9 −0.7
BERT Embeddings
Causal graph Traversal
C1 C2 C3 C4 C5
Symptom Similarity
C2 C1 C3 C5 C5
Combined Cluster Rank
Cluster Prediction
C2 C3 C1 C5 C4
Root Cause 1
Root Cause 2
Root Cause 3
...
Remediation 1
Remediation 2
Remediation 3
...

| | Metric | IS | GCN | Clust | % Gain |
|---|---|---|---|---|---|
| **Root Cause** | **Rouge-1** | 0.207 | 0.176 | 0.242 | 27.2% |
| | **Rouge-L** | 0.197 | 0.165 | 0.227 | 26.4% |
| **Remediation** | **Rouge-1** | 0.157 | 0.162 | 0.219 | 37.3% |
| | **Rouge-L** | 0.143 | 0.147 | 0.205 | 41.4% |

% Gain over two baselines

# Questions?

# Research Areas

- ML for System Reliability and Efficiency
- User Modelling for Marketing Decisions
- Multi-modal Content Generation
- Document Understanding

ML System & User Modeling

- ML Training and Inference Optimization
- Query, Compute, Storage Optimization
- Approximate Computing
- Causal Inference
- Active Learning
- Anomaly Detection
- Forecasting
- Segmentation
- Data Summarization
- NL2SQL

aaai

ESEC/FSE 2023

ICDE2023
Anaheim, CA April 3 – 7, 2023

NEURAL INFORMATION PROCESSING SYSTEMS

ACM SIGMOD

THE WEB CONFERENCE
ACM

ICML
International Conference On Machine Learning

SIGIR
Special Interest Group on Information Retrieval

MLSys

# Selected Papers: Causal Understanding of Complex Systems

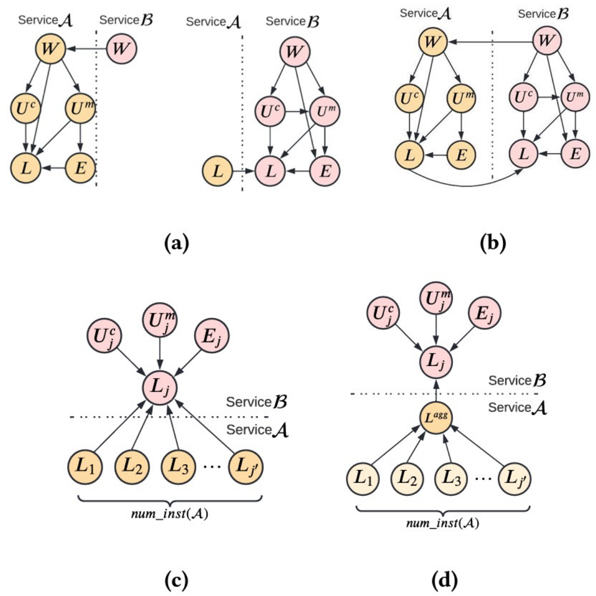



***Figure 5:** (a) shows an example estimated causal graph for individual services $\mathcal{A}$ and $\mathcal{B}$ where $\mathcal{B}$ calls $\mathcal{A}$. (b) shows the merged causal graph with error and workload merged. Figure (c) and (d) shows the parent metrics for latency of instance $j$ of $\mathcal{A}$. In (c) latency of $\mathcal{B}$ depends on metrics of $\mathcal{B}$ and latencies of all instances of $\mathcal{A}$. In (d) an aggregated latency node composed from the latencies of all instances of $\mathcal{A}$ is constructed, acting as a latent node.*

**CausIL: Causal Graph for Instance Level Microservice Data. [WWW 2023]: 2905-2915**

## Root Cause Discovery (RCD) Algorithm

1. **Hierarchical:** Split the data into small subsets and find candidate targets.
2. **Localized:** Only find the interventional targets.

**Theorem:** *Given access to a perfect conditional independence oracle, and under the causal sufficiency, and the extended faithfulness assumptions RCD returns the true root cause variables.*

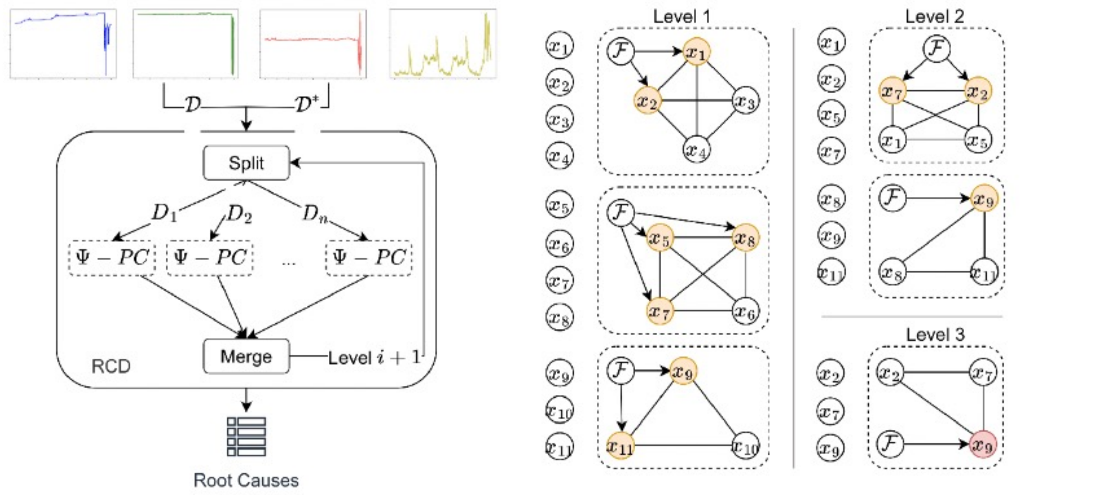



Figure 3. RCD (left) follows the divide-and-conquer approach. It first splits the data and finds the interventional targets from each subset. In the second phase, it combines the candidate root causes of all the subsets and performs the same steps recursively. The example (right) shows an execution of RCD with 11 nodes. The orange nodes are potential root causes that are carried to the next level for further processing and the red node ($x_9$) is the eventual root cause.

**Root Cause Analysis of Failures in Microservices through Causal Discovery. [NeurIPS 2022]**

# Selected Papers: Outage Prediction and Diagnosis

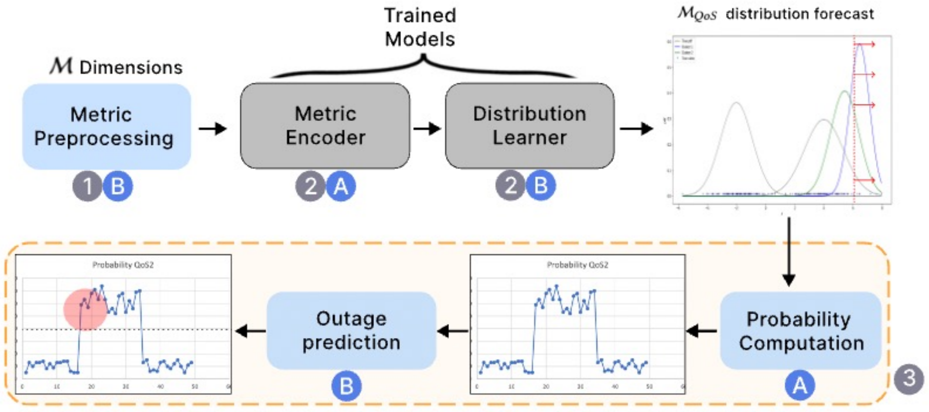



**Figure 5: Tasks performed during inference time to predict potential outages from the predicted distribution**

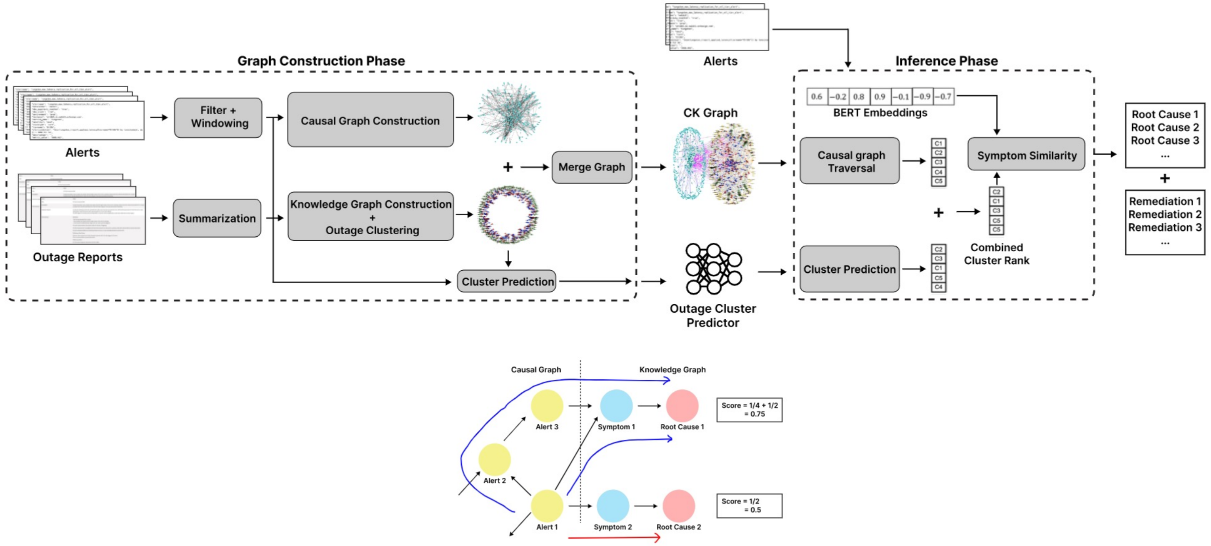
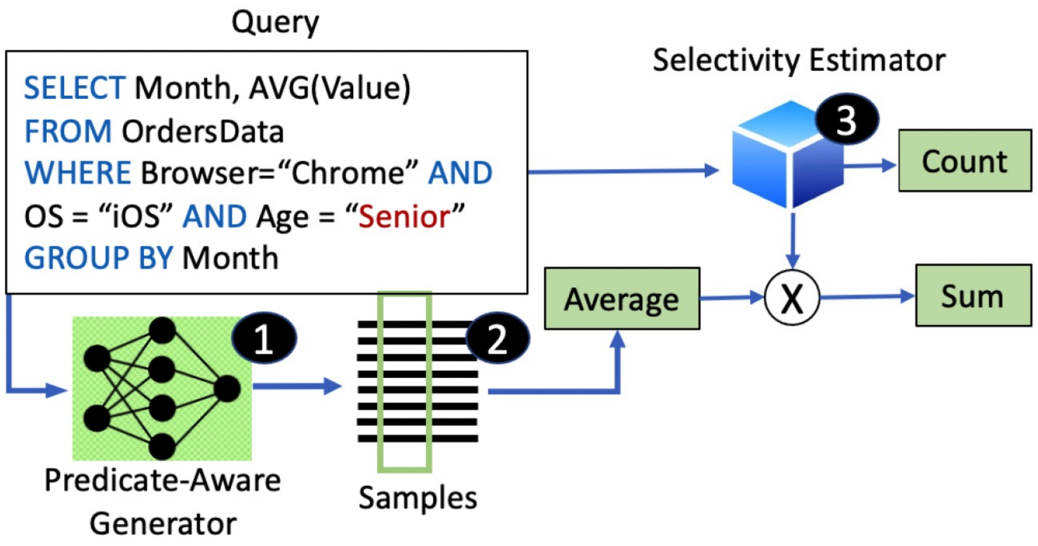


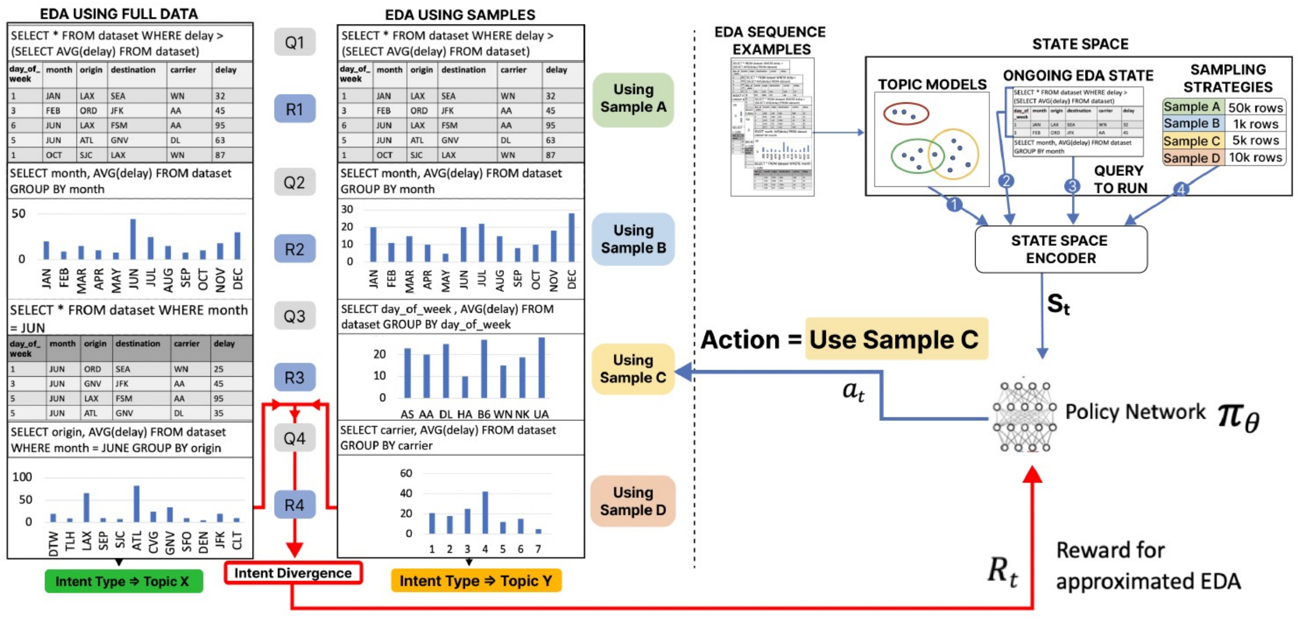

Fig. 4: *Figure shows a demonstration of the path based inference approach, where Alert 1 is fired during an outage. The inference method reaches two root causes Root Cause 1 (RC1) and Root Cause 2 (RC2) from Alert 1. There is only one 2-length path to RC2 from Alert 1 , while there are two paths to RC1, a 2-length path and a 4-length path. Hence, the score for RC1 is 0.75 while the score for RC2 is 0.5.*

**Outage-Watch: Early prediction of outages using extreme event regularizer.** ESEC/FSE 2023

**ESRO: Experience Assisted System Reliability against Outages.** ASE 2023

# Selected Papers: Approximations and ML in Big-Data Processing



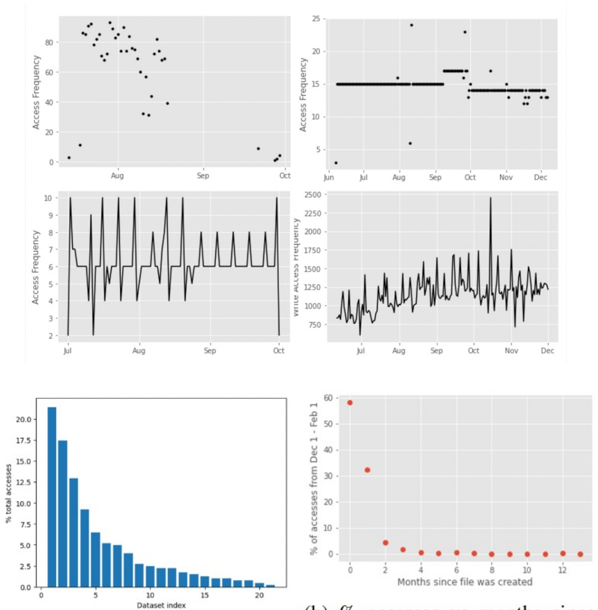

**Conditional Generative Model Based Predicate-Aware Query Approximation.** AAAI 2022: 8259-8266

**Reinforced Approximate Exploratory Data Analysis.** AAAI 2023: 7660-7669

# Selected Papers: Storage Cost Optimization

TABLE I: Cost and latency numbers for Azure [8].

| | Premium | Hot | Cool | Archive |
|---|---|---|---|---|
| Storage cost cents/GB (first 50 TB) | 15 | 2.08 | 1.52 | 0.099 |
| Read cost (cents, every 4 MB per 10k operations) | 0.182 | 0.52 | 1.3 | 650 |
| Time to first byte | Single digit ms | ms | ms | Hours |




(a) % accesses vs dataset index (b) % accesses vs months since file was created

Fig. 1: Enterprise Data access patterns

IV. OPTASSIGN: OPTIMIZING OVERALL COSTS

$$\min \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{\ell=1}^{L}[(\alpha\ C_\ell^s + \gamma\Delta_{L(P_i),\ell})\ \frac{Sp(P_n)}{R_n^k} + \beta\rho(P_n)\left(C^c\ D_n^k\ +\ C_\ell^r\frac{Sp(P_n)}{R_n^k}\right)]\ x_{n,\ell,k} \quad (1)$$

$$\text{s.t.} \sum_{\ell=1}^{L}\sum_{k=1}^{K} x_{n,\ell,k}\ =\ 1\ ,\forall\ n\ \in [N]$$

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\frac{Sp(P_n)}{R_n^k}x_{n,\ell,k}\ \le S_\ell\ ,\forall\ \ell\ \in [L]$$

$$\sum_{\ell=1}^{L}\sum_{k=1}^{K}(D_n^k\ + B_\ell)\,x_{n,\ell,k}\ \le\ T(P_n)\ ,\ n\ \in\ [N]$$

$$x_{n,\ell,k} \in \{0,1\}\ \forall n \in [N],\ \ \ell \in [L],\ \ k \in [K]$$

$$x_{n,\ell,k} = 0\ \forall n \in [I],\ \ \ell \in [L], \forall k \neq K(P_n)$$

TABLE II: % cost benefits for data across 4 customers.

| | Total Size (PB) | % Cost Benefit 2 mos | % Cost Benefit 6 mos |
|---|---|---|---|
| Customer A | 0.56 | 10.59 | **61.6** |
| Customer B | 0.45 | 8 | **53.72** |
| Customer C | 0.053 | 11.58 | **83.69** |
| Customer D | 0.085 | 9.93 | **49.6** |

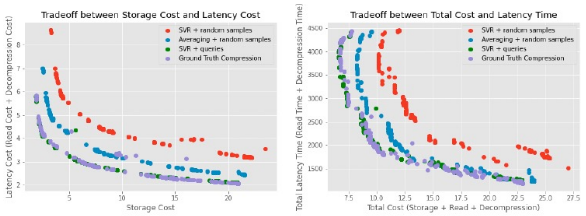
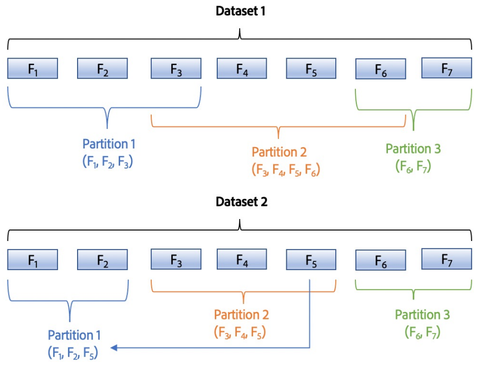
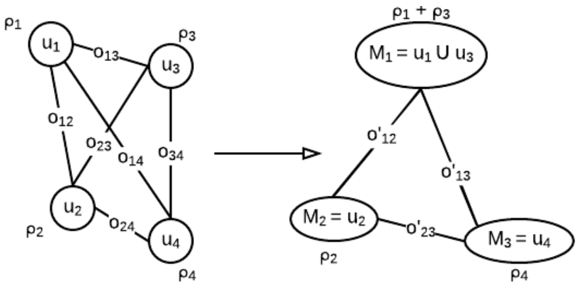



Fig. 5: Left: Latency Cost vs Storage Cost, Right: Total Cost vs Latency Time. Different tradeoff curves correspond to different compression predictors used.

VI. DATAPART: ACCESS AWARE DATA PARTITIONING




**Towards Optimizing Storage Costs on the Cloud.** ICDE 2023

# Selected Papers: Federated Learning

## Insight #2: Drift-aware Adaptive Optimizer

$\Delta^{(r)} \leftarrow \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}(w_c^{(r)} - w_g^{(r-1)})$

$m^{(r)} \leftarrow \beta_1 m^{(r-1)} + (1-\beta_1)\Delta^{(r)}$

$v^{(r)} \leftarrow \beta_2 v^{(r-1)} + (1-\beta_2)(\Delta^{(r)})^2$

$\beta_{3j} \leftarrow \frac{\left\|v_j^{(r-1)}\right\|_2}{\left\|(\Delta_j^{(r)})^2 - v_j^{(r)}\right\|_2 + \left\|v_j^{(r-1)}\right\|_2} \quad \forall j \in [n]$

$d_j^{(r)} \leftarrow \beta_{3j} d_j^{(r-1)} + (1-\beta_{3j})\left((\Delta_j^{(r)})^2 - v_j^{(r)}\right); \quad \forall j \in [n]$

$w_g^{(r)} \leftarrow w_g^{(r-1)} + \eta_g \frac{m^{(r)}}{\sqrt{v^{(r)}} - d^{(r)} + \tau}$

$$\underbrace{d^{(r)}}_{\text{higher}} = \beta_3\underbrace{(d^{(r-1)})}_{\text{lower}} + (1-\beta_3)\underbrace{((\Delta^{(r)})^2 - v^{(r)})}_{\text{higher}}$$

$$\underbrace{\eta / [\mathrm{sqrt}(v^r) - d^{(r)}]}_{\text{higher}}$$

*Faster drift adaptation (ie. convergence to the new concept)*

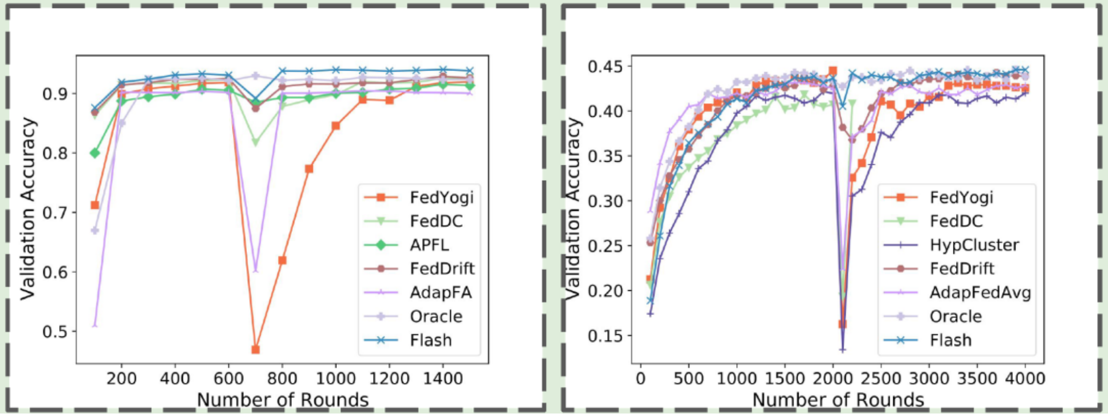


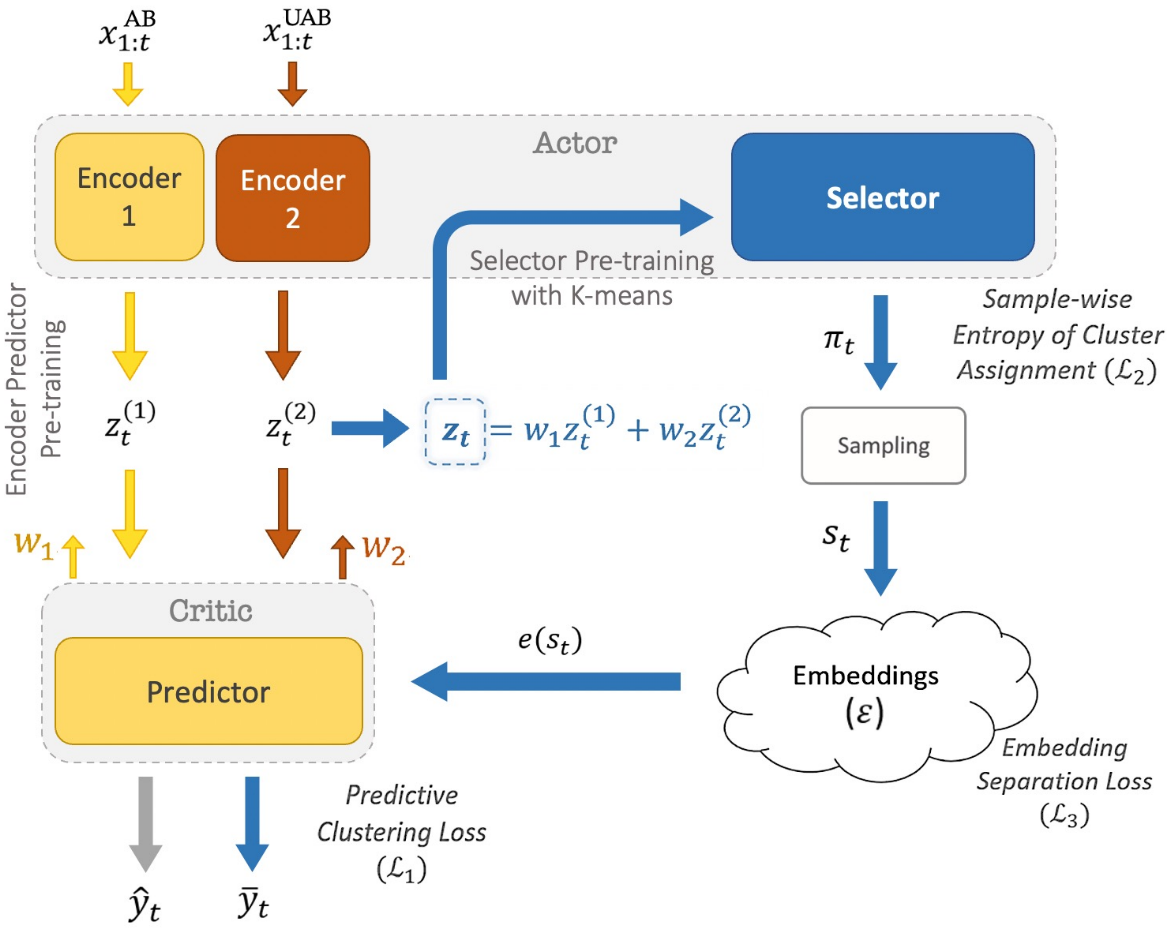

Figure: Accuracy curves with incremental drift after steady state.

**Theorem 4.2** (Convergence of FLASH). *Let assumptions C.1 to C.4 hold. Suppose the server and client learning rates satisfy* $\eta_\ell \leq \min\left[\left(\frac{|\mathcal{C}|}{30L^2E}\right)^{\frac{1}{2}}, \left(\frac{\tau}{6(B^2-1)\left[G(\beta_2+\sqrt{\beta_2})+L\eta_g\right]}\right)\right]$. *Then the iterates of Algorithm 1 for* $\eta_\ell = \Theta(1/L\sqrt{E})$, $\eta_g = \Theta(1/\sqrt{R})$, *and* $\tau = G/L$ *for* FLASH *satisfy*

$$\min_{0\le r\le R}\mathbb{E}\left\|\nabla f(w_g^{(r)})\right\|^2 \le \mathcal{O}\left(\frac{f(w_g^{(0)}) - \mathbb{E}_r[f(w_g^{(R)})]}{\sqrt{ER}} + \frac{G}{\sqrt{ER|\mathcal{C}|}}(\sigma_\ell^2 + 6E\sigma_g^2) + \frac{6L\sigma_\ell^2}{RG^2|\mathcal{C}|} + \frac{6L}{R}\right)$$

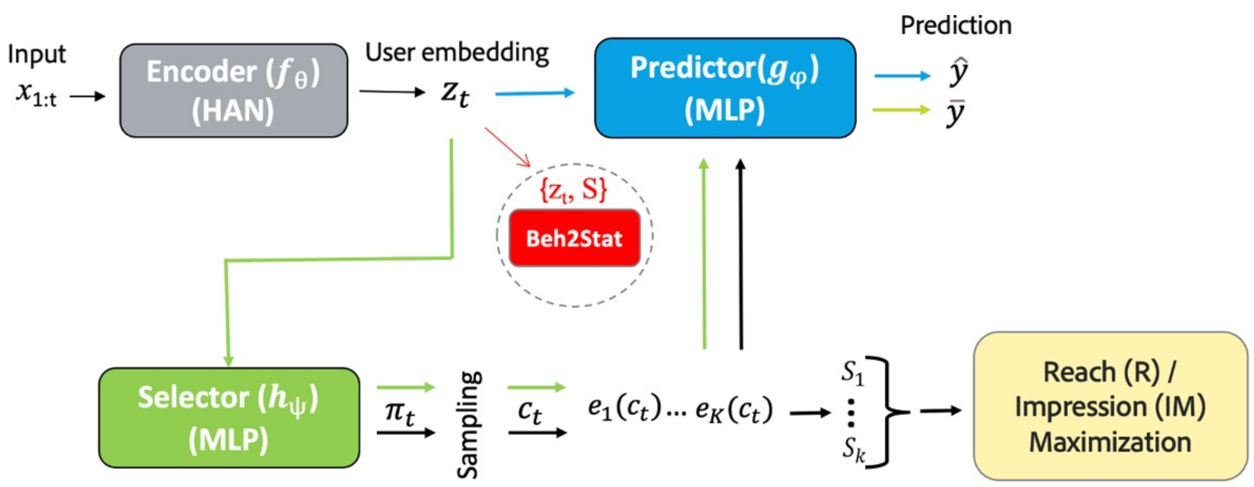
**Flash: Concept Drift Adaptation in Federated Learning.** ICML 2023

# Selected Papers: User Modeling




The Role of Unattributed Behavior Logs in Predictive User Segmentation, CIKM 2023

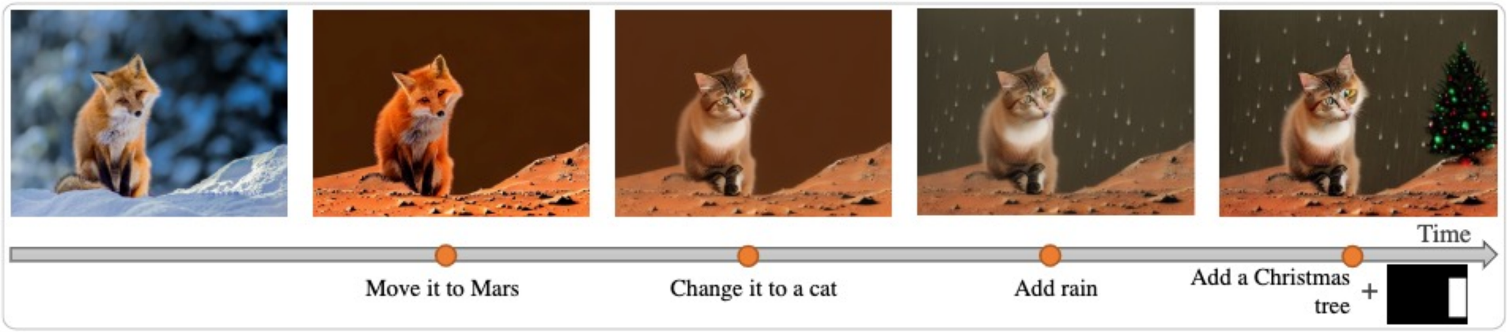
Joint Optimization of User Segmentation and Channel Delivery under Budget Constraint




Delivery Optimized Discovery in Behavioral User Segmentation under Budget Constraint, CIKM 2023

# Multimodal Content Generation

# Constrained Image Generation

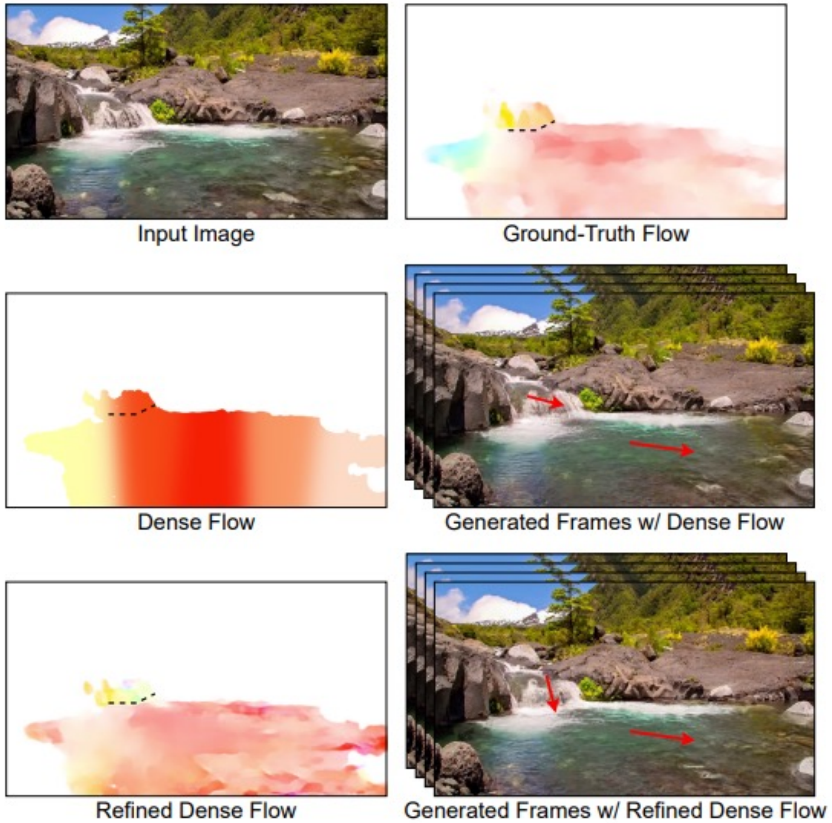




Multi-concept Generation [1]

Iterative Generation


1. Aishwarya Agarwal, Srikrishna Karanam, Joseph KJ, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan, A-STAR: Test-time Attention Segregation anad Retention for Text-to-image Synthesis, ICCV 2023

# Cinemagraph Generation

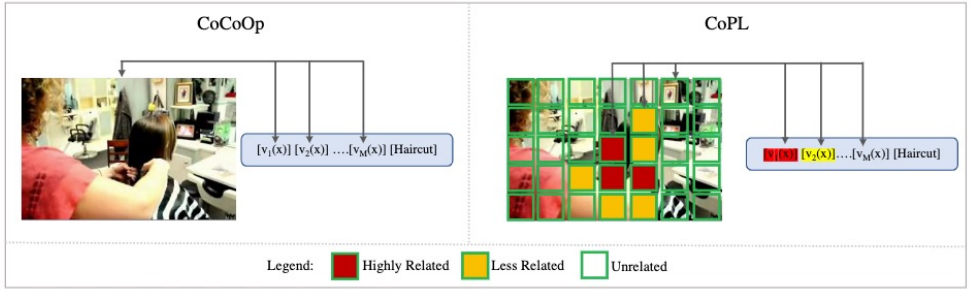


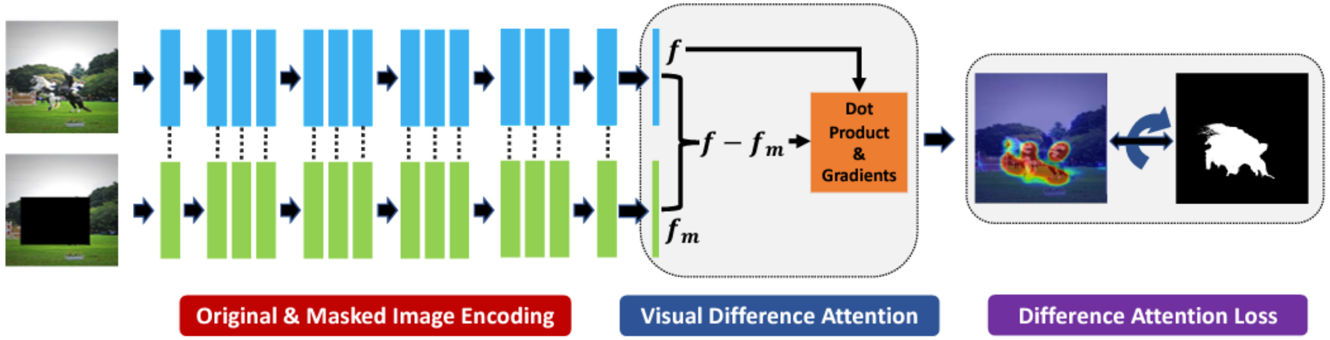

Fluid Animation [1]

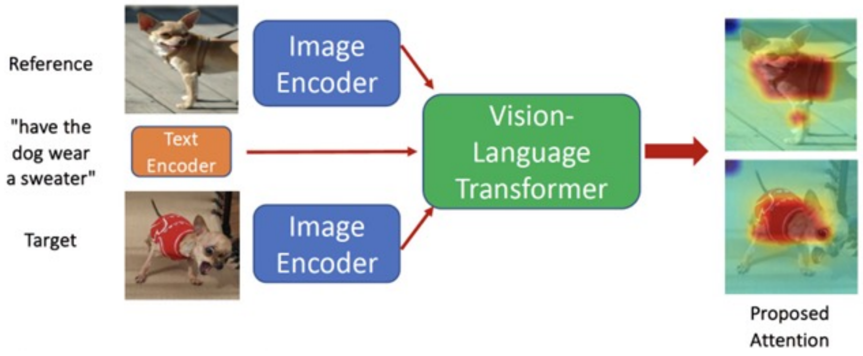
Clothing Animation [2]

1. Aniruddha Mahapatra and Kuldeep Kulkarni. "Controllable animation of fluid elements in still images." CVPR 2022.
2. Hugo Bertiche, Niloy J. Mitra, Kuldeep Kulkarni, Chun-Hao Paul Huang, Tuanfeng Y. Wang, Meysam Madadi, Sergio Escalera and Duygu Ceylan, "Blowing in the Wind: CycleNet for Human Cinemagraphs from Still Images", CVPR 2023

# Multimodal Representation & Grounding



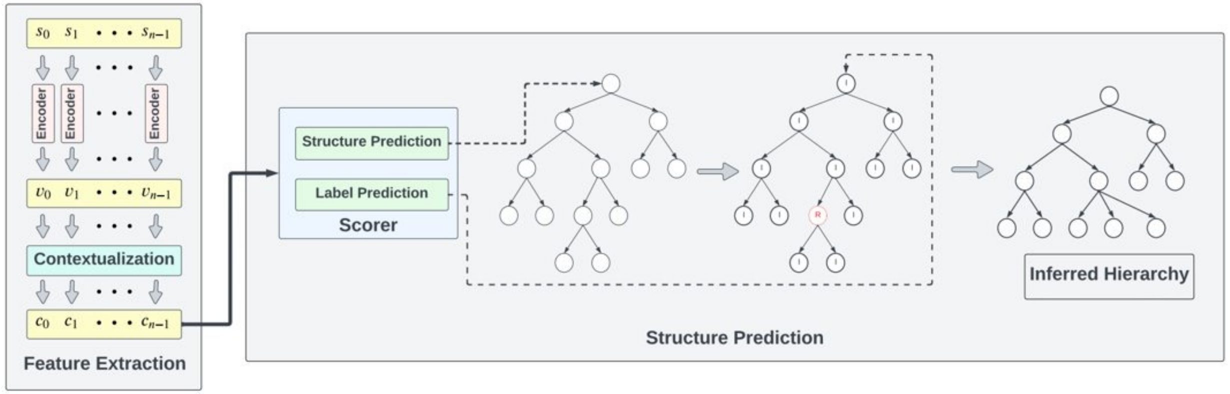

Multimodal Representation [1,2]

Multimodal Grounding [3]

1. Aishwarya Agarwal, Srikrishna Karanam, and Balaji Vasan Srinivasan, Learning with Difference Attention for Visually Grounded Self-supervised Representations, arXiv 2023, under review

# Natural Language Processing

# LLMs for Document Understanding & Consumption



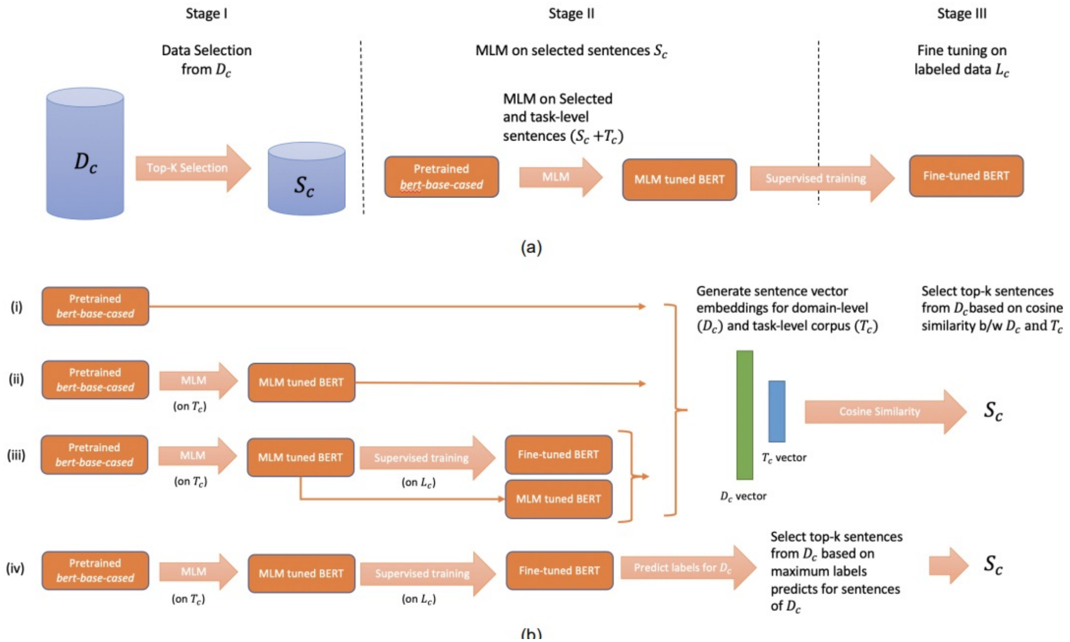

Document Navigation [2]

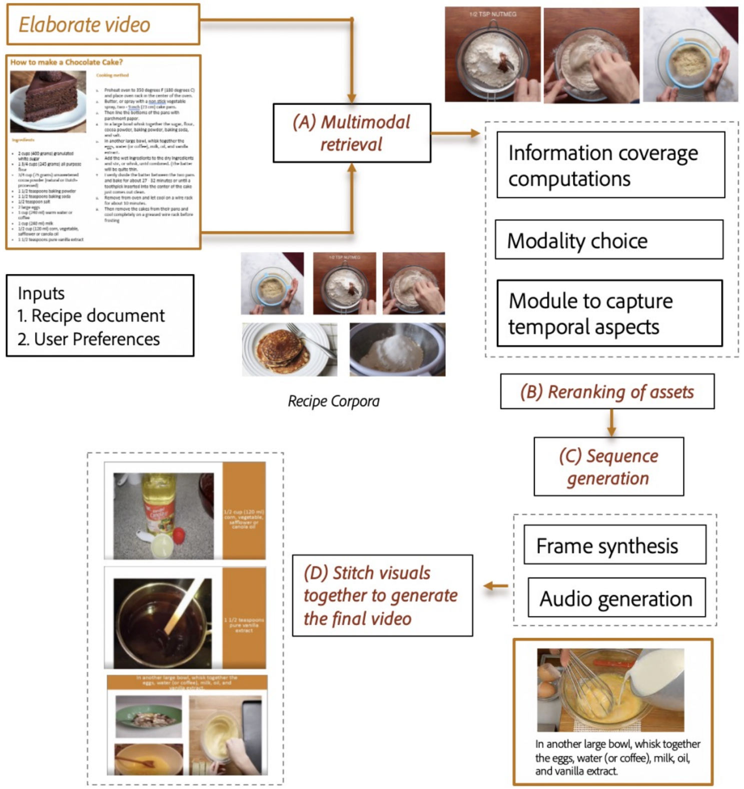
Document Segmentation [1]


(a) Input Document (b) Default Table of Contents (c) DynamicToC Personas (d) DynamicToC Table of Contents


1. Inderjeet Nair, Aparna Garimella, Balaji Vasan Srinivasan, Natwar Modani, Niyati Chhaya, Srikrishna Karanam, Sumit Shekhar; A Neural CRF-based Hierarchical Approach for Linear Text Segmentation; EACL 2023
2. Maheshwari, Himanshu, Nethraa Sivakumar, Shelly Jain, Tanvi Karandikar, Vinay Aggarwal, Navita Goyal, and Sumit Shekhar. "DYNAMICTOC: Persona-based Table of Contents for Consumption of Long Documents." NAACL 2022.

# Document Transformations




Cross Modal Transformation [WACV 2023]