Data-Centric AI

Powering the Machine Learning Lifecycle from Preparation to Reliable Inference

Prof. Lei CHEN

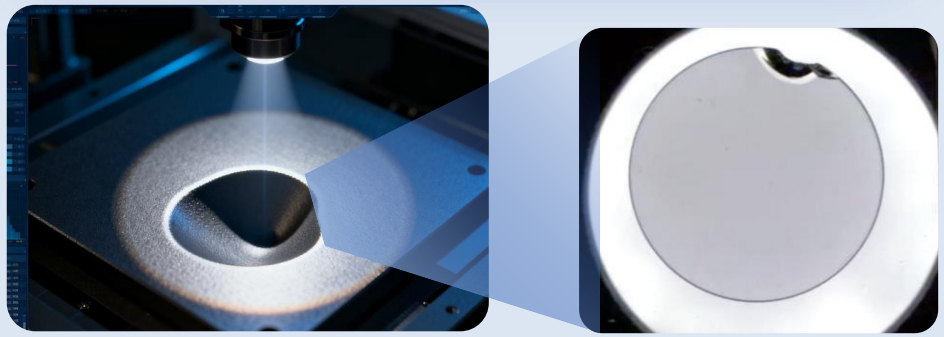
Data Science and Analytics (DSA) Thrust - Information Hub
The Hong Kong University of Science and Technology (GZ)

Outline

Background: From Model-Centric AI to Data-Centric AI

From the view of effectiveness, a purely model-centric approach is limited.

BMW Group
Surface Defect Detection



MCAI

- Data
- Model
  - Architecture
  - Hyperparameter
  - Tuning

Performance Limitation

83%

DCAI

- Data
  - Preparation
  - Labeling
  - Segmentation
- Model

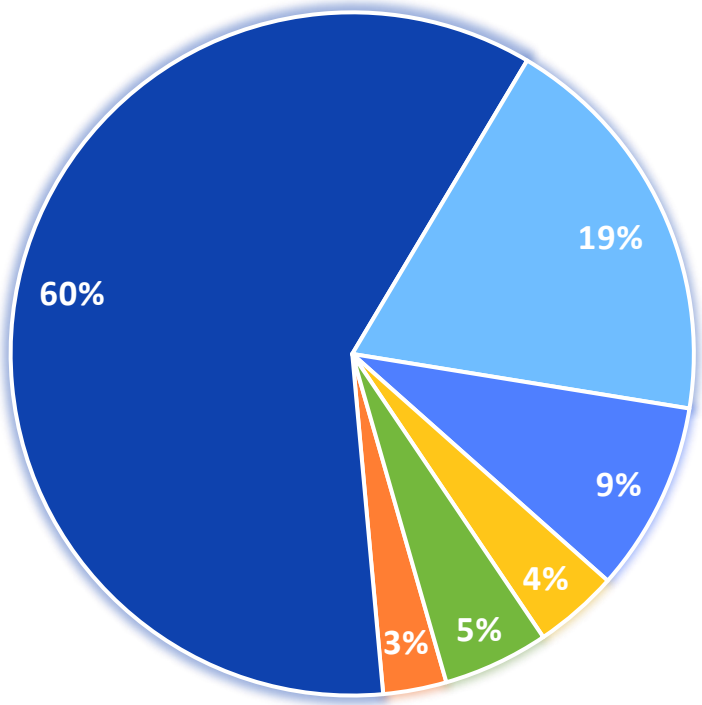
Performance Breakthrough

93%

Nieberl, M. et.al. Enhancing the Performance of Computer Vision Systems in Industry: A Comparative Evaluation Between Data-Centric and Model-Centric Artificial Intelligence. Electronics 2025

Background: From Model-Centric AI to Data-Centric AI

Data issues still account for the majority of an AI engineer's workload.



- Building training sets：3%
- Cleaning and organizing data：60%
- Collecting data sets：19%
- Mining data for patterns：9%
- Refining algorithms：4%
- Other：5%

80%

of ML users' time/effort (often more) spent on data issues !"

Arun Kumar. 2021. Automation of Data Prep, ML, and Data Science: New Cure or Snake Oil? In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21).

Background: From Model-Centric AI to Data-Centric AI

Feeding subpar data into AI models leads to severe financial loss.

Bad Data and Trust Erosion

42% Experience data inaccuracies/hallucinations

40% Suffer from data biases that may stem from incomplete data

40% Struggle with slow access to data, impacting the relevancy of insight

Investment and Data Quality

13% Average global annual revenue invested in AI/ ML model development in the next 1-2 years.

$406M Average global annual revenueloss due to underperforming AImodels from inaccurate or low-quality data.

(based on respondents from organizations with anaverage global annual revenue of $5.6 billion USD)

Fivetran + Vanson Bourne report: AI in 2024

Overview: Data-Centric AI Across the ML Lifecycle

Machine Learning Lifecycle

01 Data Preparation

Raw data → learnable signals

Challenge 1: Data Preparation

Why important

Determines data quality, robustness, and generalization.

02 Model Training

Curated data → efficient learning

Challenge 2: Model Training

Why important

Reduces training cost, accelerates convergence, and improves model capability.

03 Model Inference

Runtime data flow → efficient serving

Challenge 3: Cache and Batch Techniques

Why important

Improves latency,throughput, serving cost, and scalability.

04 Reliable Generation

Grounding data → reliable outputs

Challenge 4: Reliable Generation

Why important

Reduces latency, ensuring reliability and factual correctness.

Challenge 1: Data Preparation

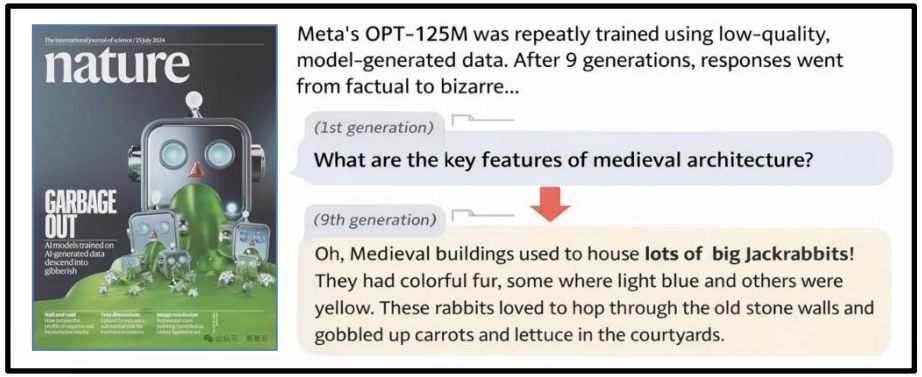
Case 1

Amazon scrapped 'sexist AI' tool

The model inherited historical bias from the training data.

Case 2

Nature - Model Collapse with Low-Quality Synthetic Data



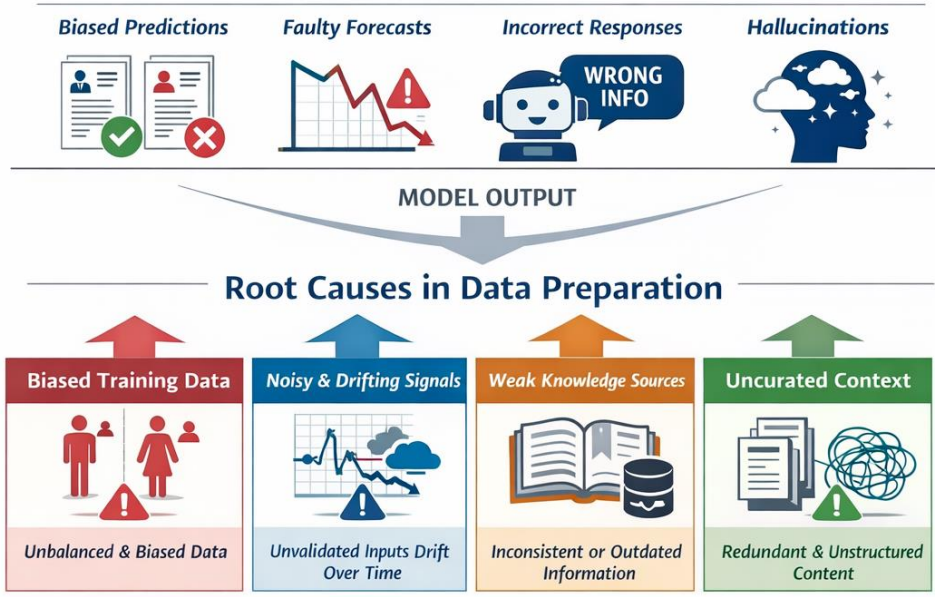
Poor-quality, model-generated data caused factual drift.

Challenge 1: Data Preparation

Low-Quality Data Leads to Poor Models

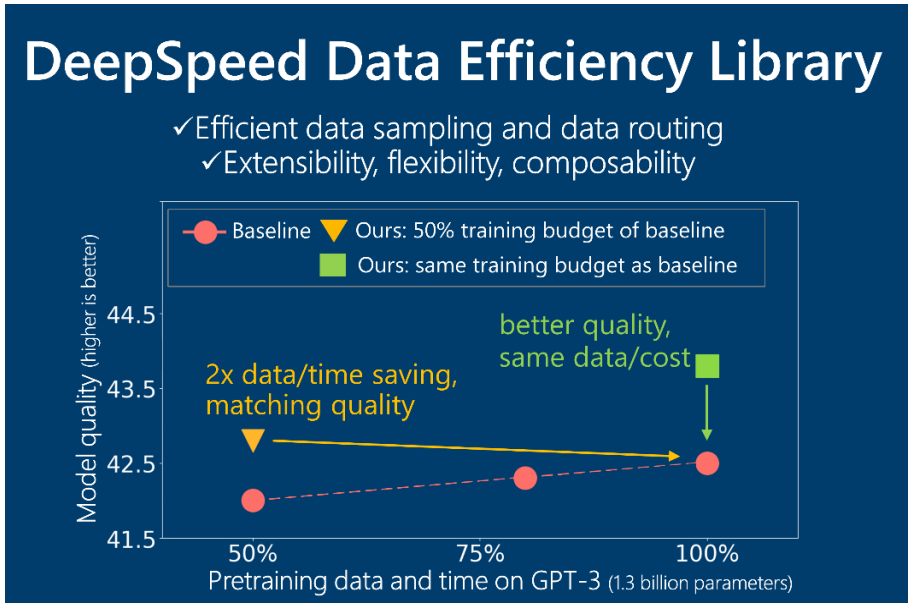
Bottlenecks

The quality of model output is fundamentally limited by data preparation.



Apt-Serve Insights

High quality input data makes more efficiency & accuracy.



Challenge 2: Model Training in LLM Era

Key Challenge: Massive Unstructured Data Processing

How to handle massive, unstructured, high-dimensional multimodal data?

Structured single modality

One Picture

↓

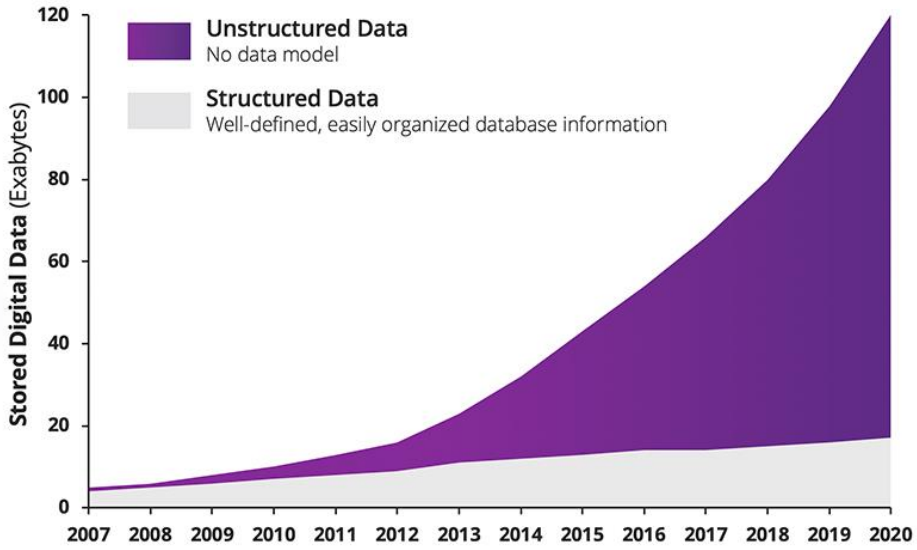
1000+ tokens

↓

30+ training hours

InternVL2-8B MLLM training

Unstructed multi-modality

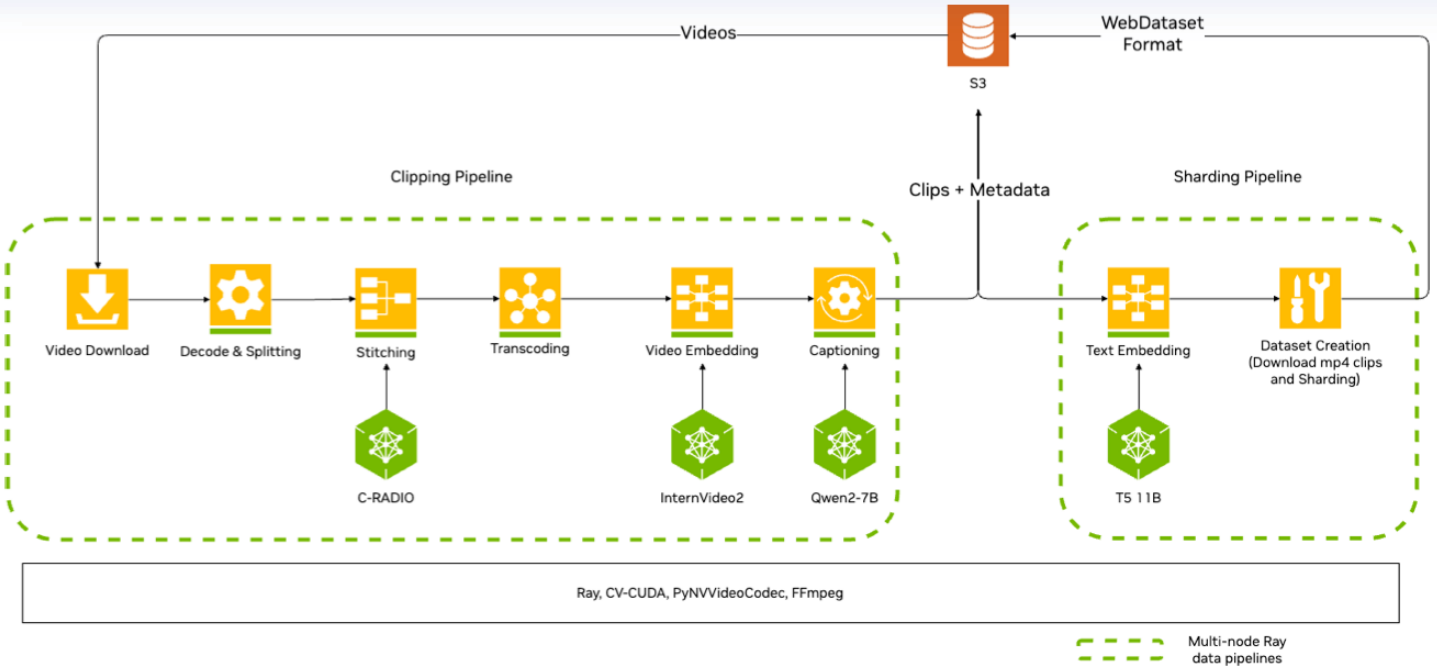


Chen, Zhe, et al. "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024.

Challenge 2: Model Training in LLM Era

Storage Efficiency Bottlenecks

How to break the storage efficiency barrier for petabyte-scale training data?



Training large model requires

100TB

scale high-quality data.

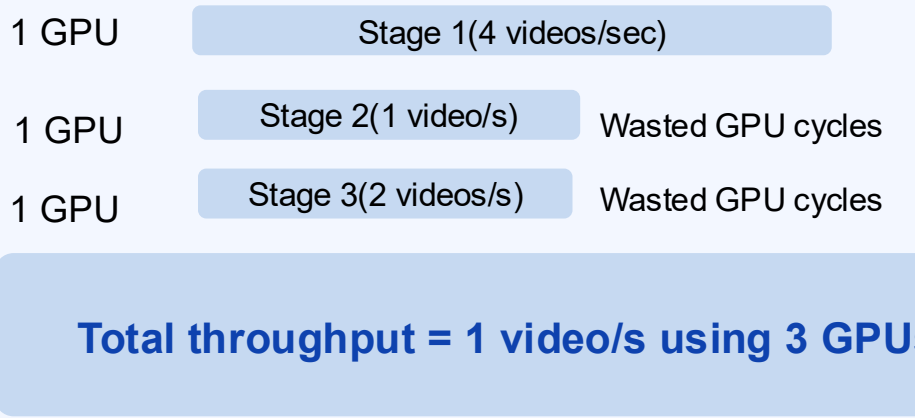
Training Video Foundation Models with NVIDIA NeMo, NVIDIA

Challenge 2: Model Training in LLM Era

Computational Resource Scaling

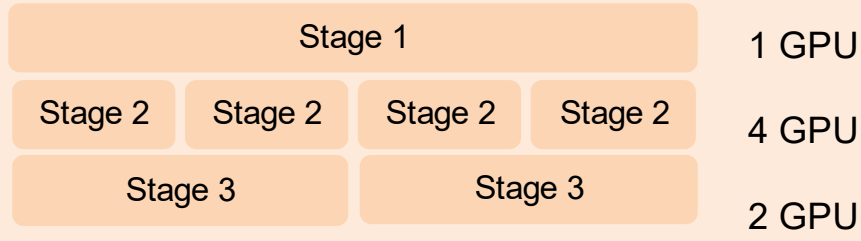
How to to improve resource utilization in large-scale training clusters?

Before



Total throughput = 1 video/s using 3 GPUs

After



Total throughput = 4 video/s using 7 GPUs

1.7X improvement

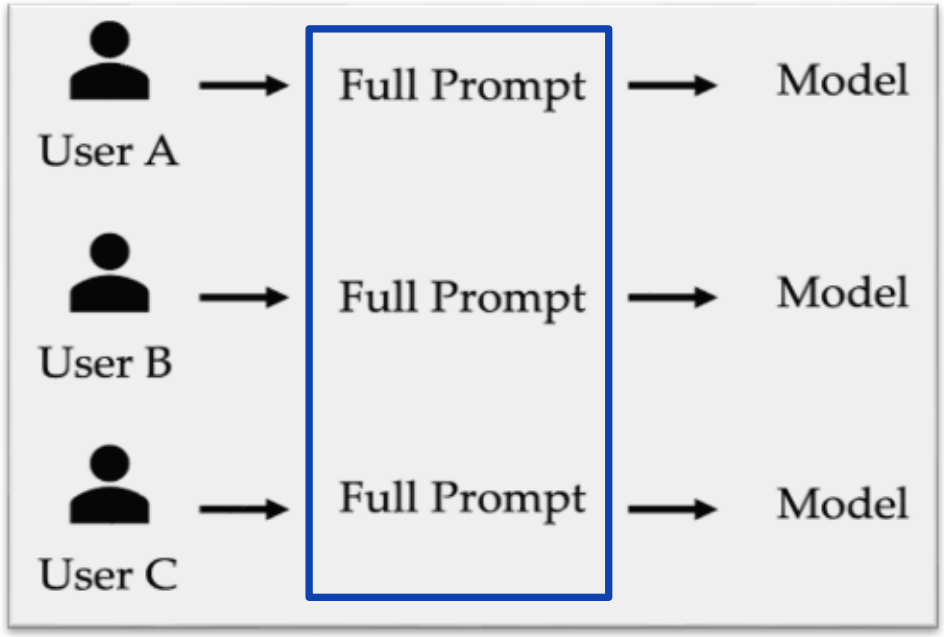
Different Inference Stages have different compute requirements

Training Video Foundation Models with NVIDIA NeMo, NVIDIA

Challenge 3: Cache and Batch Techniques

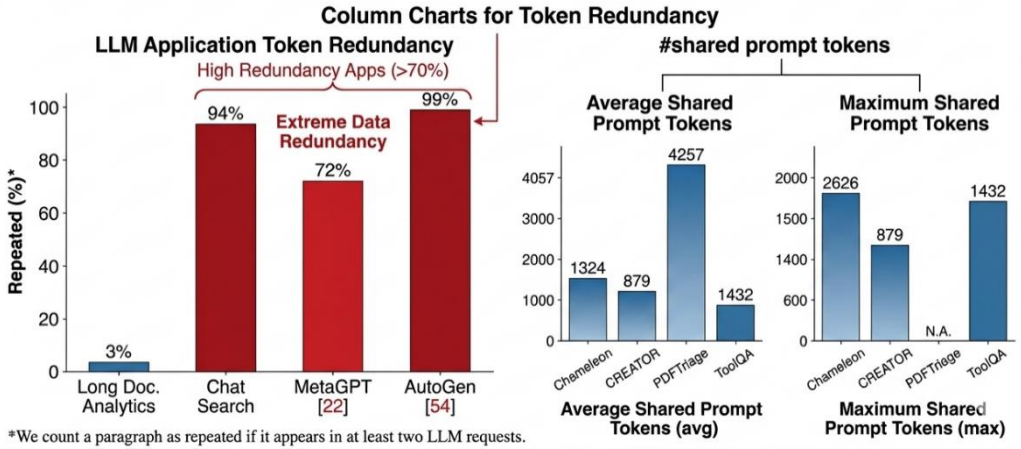
Key Challenge: Data Redundancy in Inference

Naïve Inference



Repeated Tokens Processing

Examples



Over **94%** Repeated

Parrot: Efficient Serving of LLM-based Applications with Semantic Variable

ChunkAttention: Efficient Self-Attention with Prefix-Aware KV Cache and Two-Phase Partition

Challenge 3: Cache and Batch Techniques

Input-level: Redundant Shared Tokens

Chatbot models process sequences with a large shared contents, such as prefix tokens.

Shared Prefix Setting

Shared Prefix

```
You are ChatGPT, a large language model
trained by OpenAI, based on the GPT-4
architecture.
Knowledge cutoff: 2023-04
Current date: 2023-11-16

Image input capabilities: Enabled

When you send a message containing
Python code to python, it will be
executed in a stateful Jupyter notebook
enrivonment. Python will respond...
```

Unique Suffixes

```
Hi, can you write a...
```

```
Tell me a funny...
```

```
Who is Alan Turing?
```

```
Debug this Python...
```

```
Ignore all previous...
```

Shared system prompts can exceed

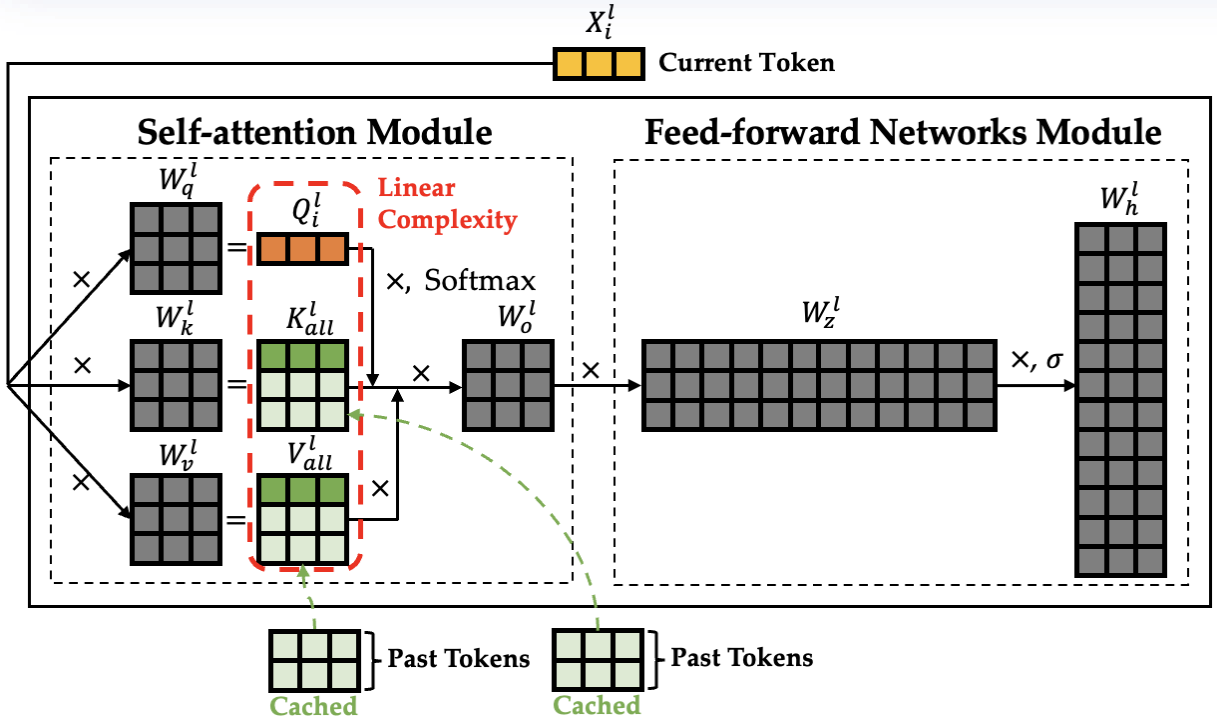
1K tokens

System	Usage of Prompt	#shared prompt tokens	
		avg	max
Chameleon	Tools definition and examples [1]	1324	2626
CREATOR	CoT examples [2]	879	2492
PDFTriage	PDF document metadata	4257	N.A.
ToolQA	Tools definition and examples [3]	1432	1432

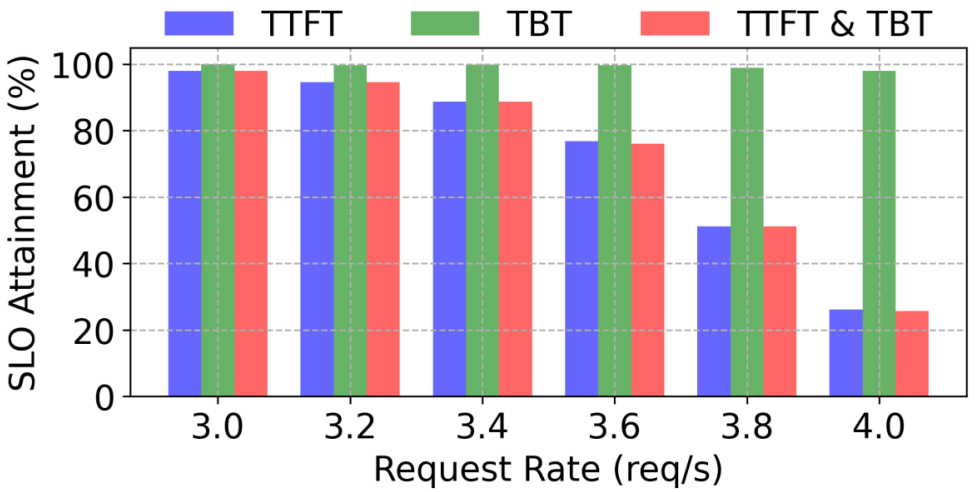
Challenge 3: Cache and Batch Techniques

Execution-level: Redundant KV Access

The inability to maintain the TTFT SLO attainment by the existing systems.



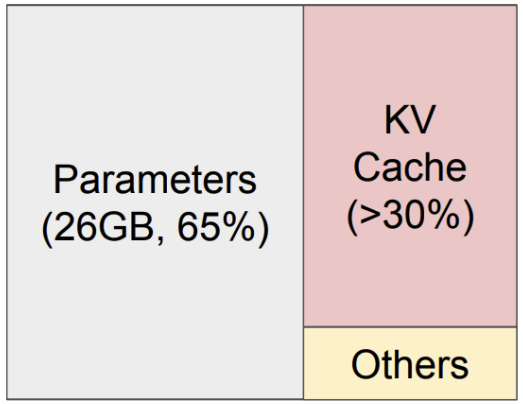
Overwhelming KV cache (**>12GB** for a request of length 8192 over a 30B model)



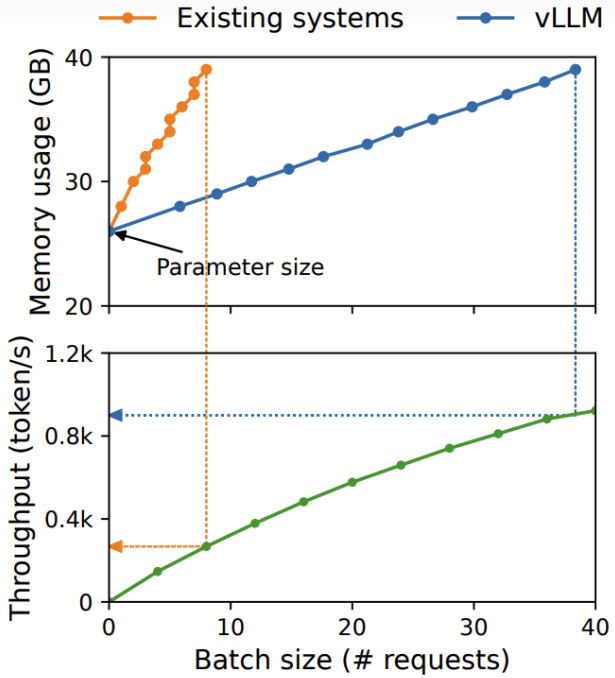
Challenge 3: Cache and Batch Techniques

System-level: Memory Scheduling Constraints

May not have contiguous memory to batch shared prefix

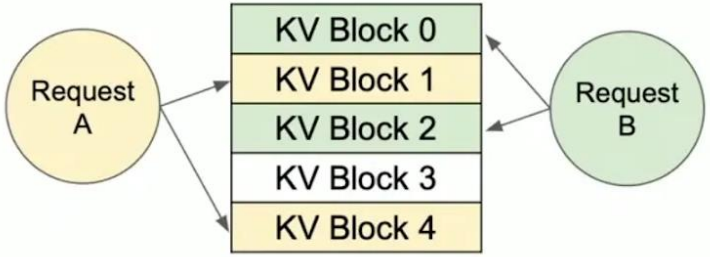


NVIDIA A100 40GB



Effective memory utilization can be as low as **20%**

Memory management in vLLM



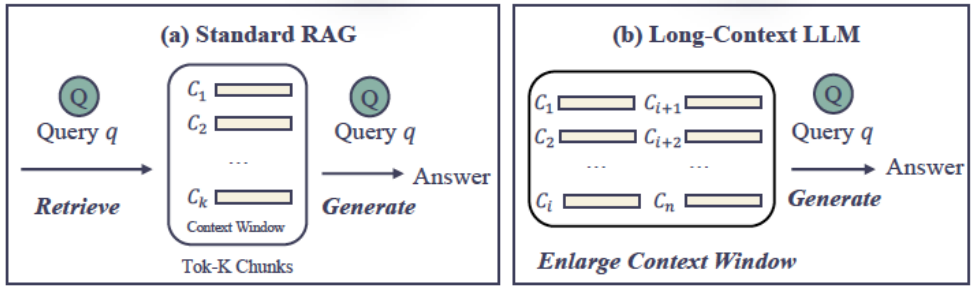
KV Cache

Challenge 4: Reliable Generation

Key Challenge: Unreliable Outputs due to Missing or Misused Data

Naïve Inference

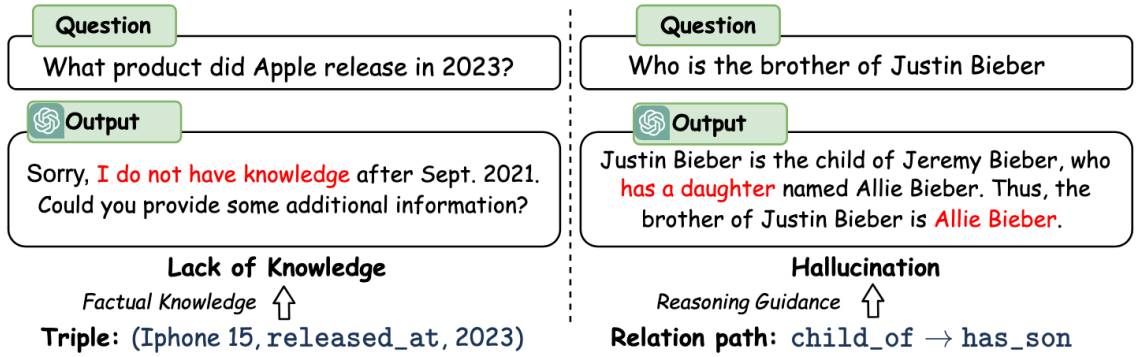
User Query → (Retrieve) Model → Answer



- no grounding
- no verification
- no uncertainty awareness

Hallucination & Unreliability

Examples



out-of-date knowledge, factual error, and opaque decision-making

Skyline Retrieval meets Set-Cover Chunk Merging: A Cost-Effective RAG-Sketch for Long-Context LLM QA

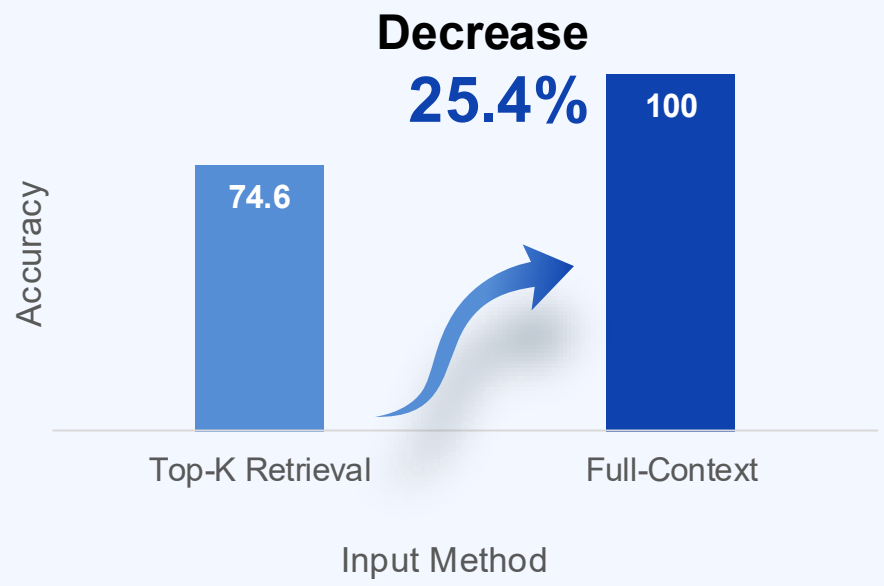
VoG: Enhancing LLM Reasoning through Stepwise Verification on Knowledge Graphs

Challenge 4: Reliable Generation

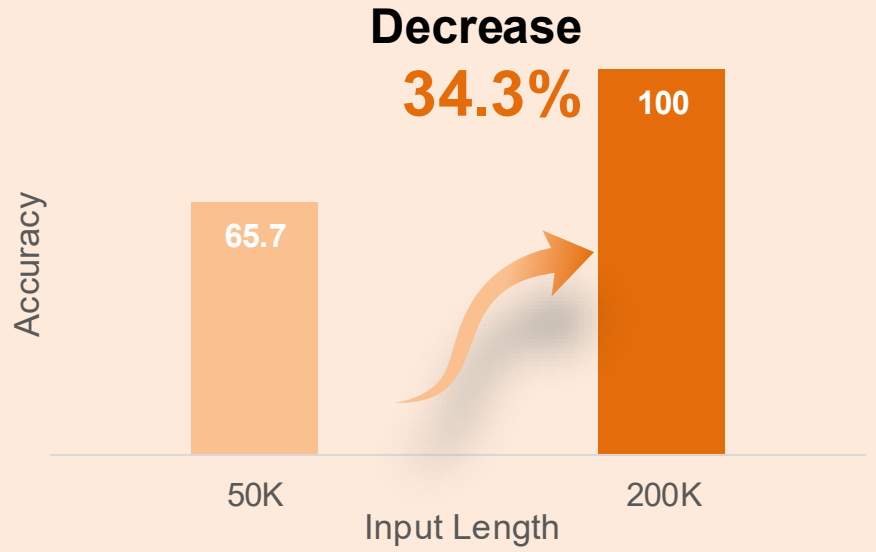
Input-level: Data Selection Problem

Hard to balance comprehensive retrieval with high token costs.

Limited Retrieved Information Issue



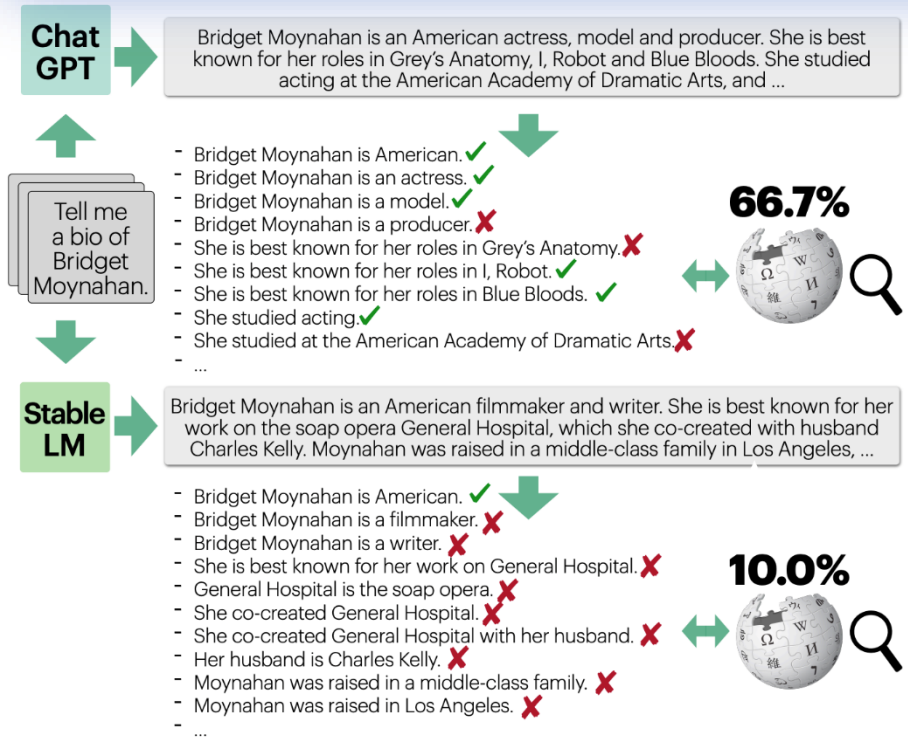
Truncation Issue

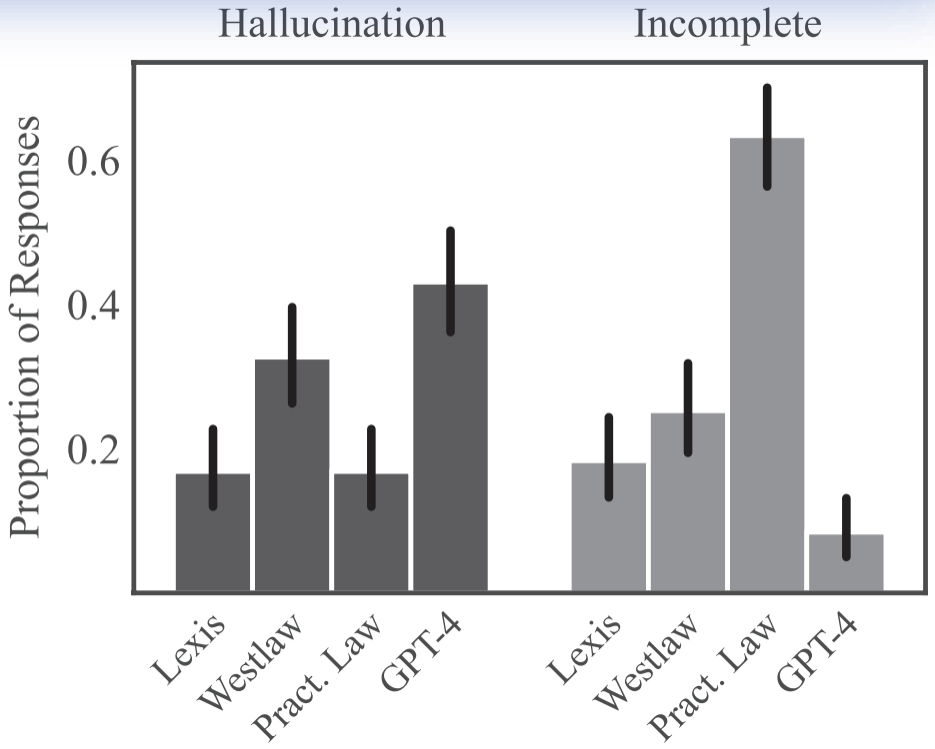


Challenge 4: Reliable Generation

Execution-level: Misuse of Retrieved Data

Generations often contain many unsupported pieces of information.





FACTSCORE: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation
Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools

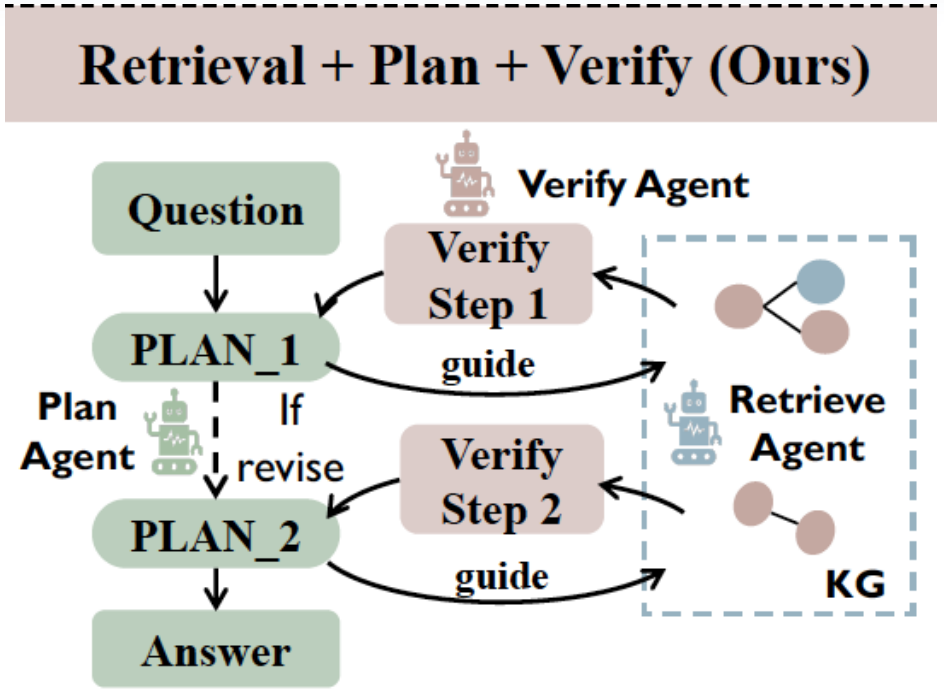
Challenge 4: Reliable Generation

System-level: Lack of Data Validation

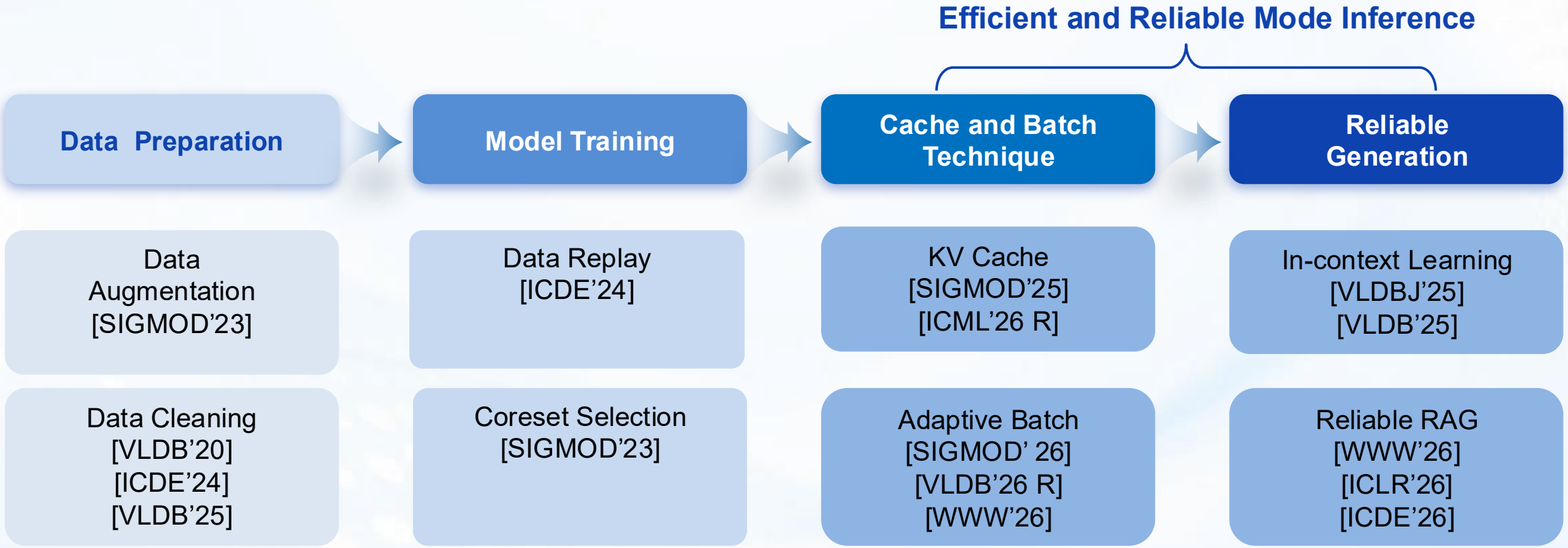
Existing works lack an explicit verification mechanism.

Table 4: Comparison of recent KG-enhanced LLM reasoning methods from three perspectives.

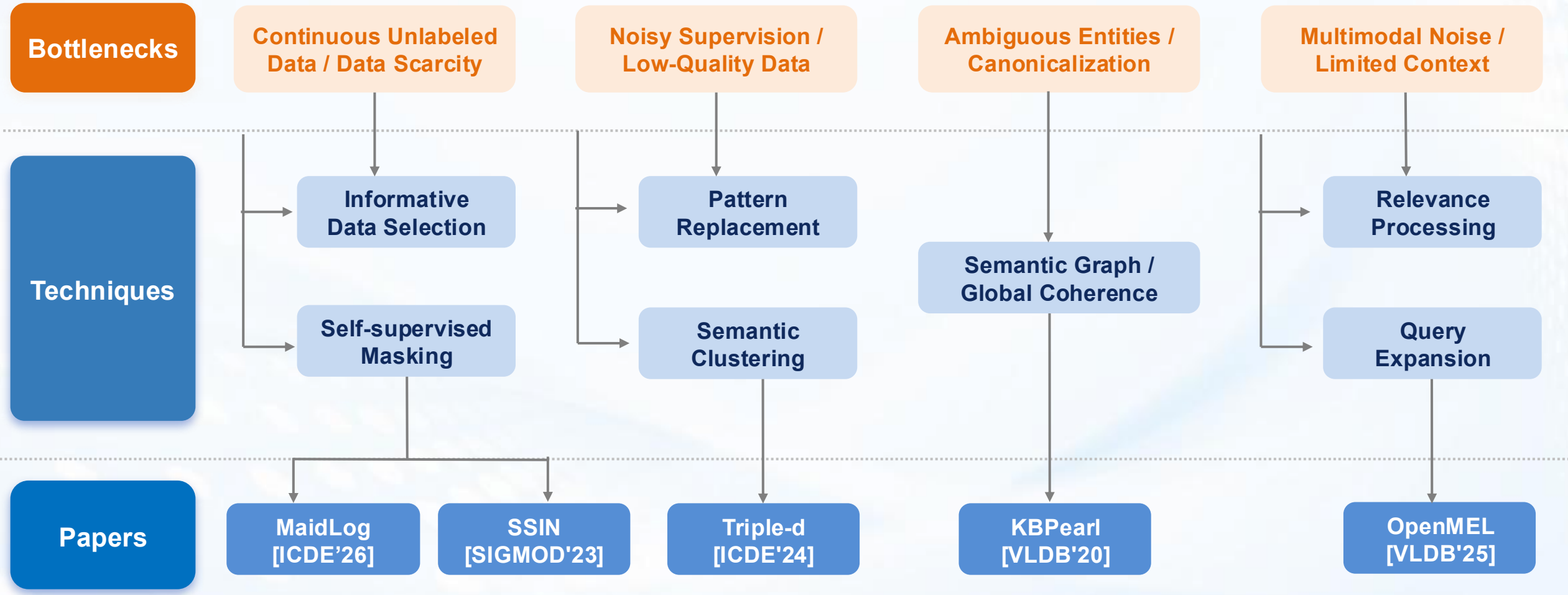
Method	Plan	Retrieve	Verify
RoG (Luo et al., 2024b)	✓	✗	✗
KnowGPT (Zhang et al., 2024)	✗	✓	✗
KG-Agent (Zhao et al., 2024)	✓	✓	✗
KD-COT (Wang et al., 2023b)	✗	✓	✓
ToG (SUN et al., 2024)	✗	✓	✗
PoG (Chen et al., 2024)	✓	✓	✗
VoG (Ours)	✓	✓	✓



Overview of Our Research



Data Preparation



Multimodal Noise [VLDB'25]: Key Insights

Problem of Multimodal Entity Linking: link the multimodal mention to the most similar multimodal entity in the KB.

Key Insight 1

Improve the input multimodal data quality through **three-level relevance** processing.

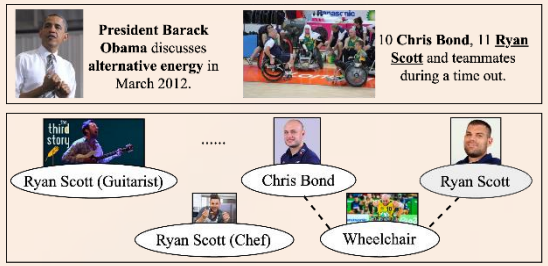
Key Insight 2

mention-to-entity

10 Chris Bond, 11 **Ryan Scott** and teammates...

- Ryan Scott (Wheelchair)
- Ryan Scott (Chef)

entity-to-entity helps



Three-level Relevance

High Relevance

President Barack Obama discusses **alternative energy** in March 2012.

Middle Relevance

10 **Chris Bond** 11 **Ryan Scott** and teammates during a time out.

Low Relevance

President Barack Obama discusses **alternative energy** in March 2012.

Multimodal Noise [VLDB'25]: Framework Overview

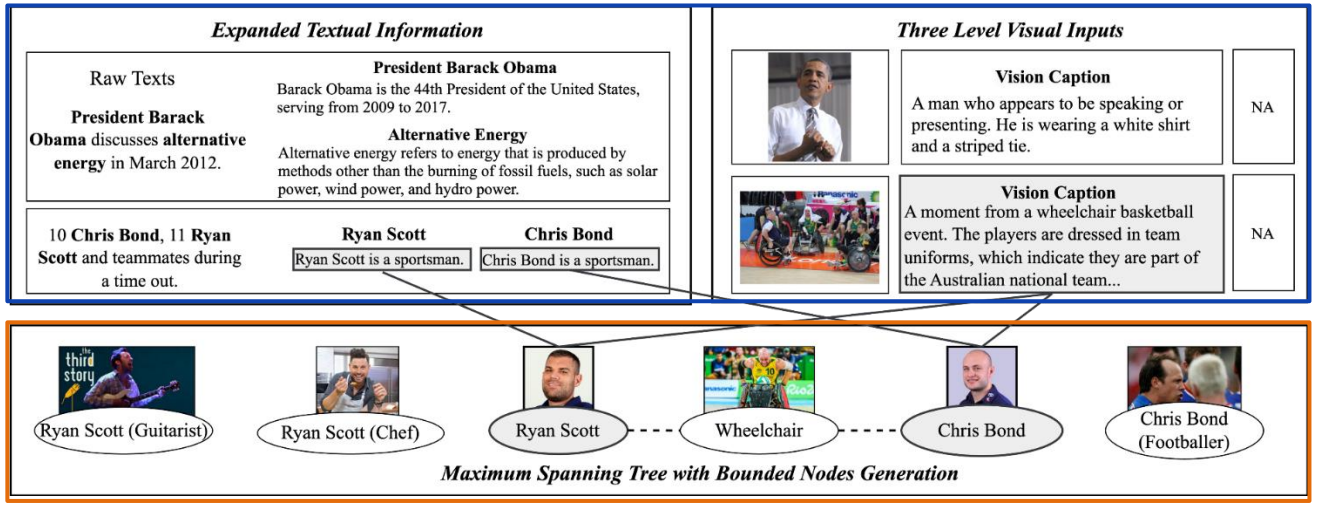
OpenMEL links multimodal mentions through noise-free query expansion and global coherence resolution.

Module 1:
Noise-free Expanded Queries Generation

Remove noise in visual inputs and expand limited context in textual inputs

Module 2:
Global Coherence Resolution

Establish mention-entity and entity-entity relationships for linking



Multimodal Noise [VLDB'25]: Queries Generation

I. Noise Removal

We process **three-level** visual inputs

- ∅: without image information
- image caption T_σ
- original image V_m

Visual inputs: Generate General Vision Caption

What is in the image?

The image depicts a moment from a wheelchair basketball event. The players are dressed in team uniforms, which indicate they are part of the Australian national team.

Visual inputs: Generate Specific Vision Caption

The image is about {Hari}. Also give you related context that {Hari in 2007}. Give the possible occupation of the person in the figure WITHOUT explanations. Just generate in one sentence.

The person in the figure is likely to be a public figure or celebrity from 2007, possibly involved in entertainment or sports.

II. Context Expansion

To address **Limited Context T_m**, we **add contextual information** with general knowledge by LLMs.

Textual inputs: Add Contextual Information

You will be given a text and a specific mention. We will further link the mention to an entity in the existing knowledge base. Please provide more information about this mention which will help our further linking. Please give the information following the example. Given text: A stamp issued by Russia for the Paralympic Games held in Sochi in 2014. mention: Russian. Information you should generated: Russia is a country spanning Europe and Asia. Second example. Given text - Bobby Fischer in his younger days. mention: Fischer. The information you should generated is Bobby Fische is an American chess player and writer. DO NOT give any other additional sentence except the information.

Sure! I'd happy to help you.

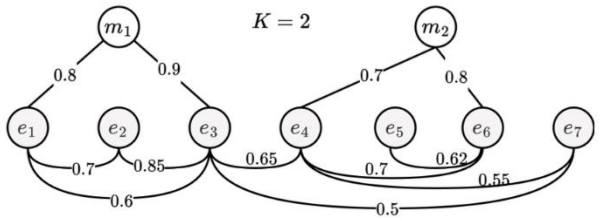
Full Template
Question: Raw Text is {Textual input}, the given mentino is {Mention}.

Example
Raw text is President Barack Obama discusses alternative energy in March 2012, the given mention is alternative energy.

Alternative energy refers to energy that is produced by methods other than the burning of fossil fuels, such as solar power, wind power, and hydro power.

Multimodal Noise [VLDB'25]: Coherence Resolution

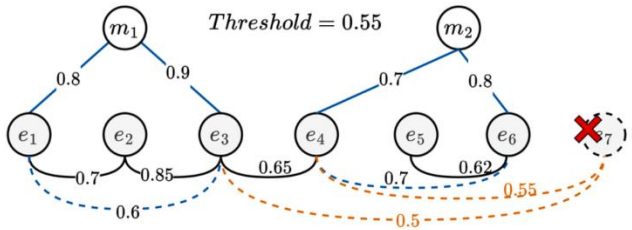
I. Coherence Data Structure



(a) Tree Cover Construction.

- The tree cover is a weighted undirected heterogeneous graph.
- The root of each tree is a specific mention awaiting linking.
- The rest tree node belongs to the entity

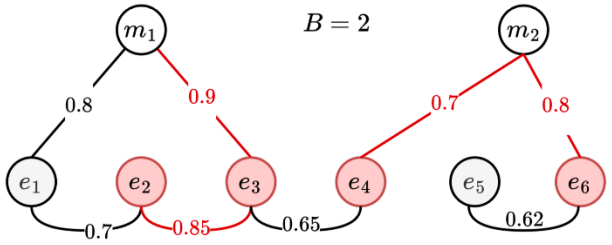
II. Edge Pruning



(b) Edge Pruning.

- Reduce the complexity of further linking
- Set a pre-defined threshold to first filter the low coherence edges
- Remove cycles associated with the tree root

III. Approximate Resolution



(c) Maximum Spanning Tree Generation.

Approximation ratio: 1-1/e

- Define the linking problem in tree cover structure as Maximum Spanning Tree (MST) problem
- Start Node: root mention m
- Greedily select the entity which will not form a cycle while bring the maximum weighted edge

Multimodal Noise [VLDB'25]: Experimental Performance

- OpenMEL demonstrates competitive performance.
- Directly employ visual information with noise leads to limited performance.

C	M	Methods	WikiMEL			RichpediaMEL			WikiDiverse		
			HIT@1	HIT@5	HIT@10	HIT@1	HIT@5	HIT@10	HIT@1	HIT@5	HIT@10
S	T	ARNN	32.0%	45.8%	56.6%	31.2%	39.3%	45.9%	22.4%	50.5%	68.4%
	T+V	DZMNED	34.7%	53.9%	58.1%	32.4%	43.7%	48.2%	-	39.1%	-
	T+V	JMEL	31.3%	49.4%	57.9%	29.6%	42.3%	46.6%	21.9%	54.5%	69.9%
	T+V	MEL-HI	38.6%	55.1%	65.2%	34.9%	43.1%	50.6%	45.7%	76.5%	88.6%
	T+V	GHMFC	43.6%	64.0%	74.4%	38.7%	50.9%	58.5%	46.0%	77.5%	88.9%
	T+V	DWE	**44.7%**	**65.9%**	**80.8%**	**67.6%**	**97.1%**	**98.6%**	**47.5%**	**81.3%**	**92.0%**
U	T	BERT	31.7%	48.8%	57.8%	31.6%	42.0%	47.6%	22.2%	53.8%	59.8%
	T	BERT (Aligned)*	40.5%	54.1%	61.5%	37.6%	48.2%	50.4%	30.1%	59.9%	63.4%
	T+V	CLIP*	40.7%	56.0%	64.6%	38.1%	54.5%	62.4%	34.4%	59.7%	62.2%
	T+V	CLIP (Aligned)*	45.7%	64.2%	71.3%	41.4%	57.8%	65.3%	41.6%	64.8%	68.5%
	T+V	BM*	33.2%	50.7%	57.5%	45.1%	62.3%	69.9%	28.8%	48.8%	58.1%
	T+V	BM (Aligned)*	57.8%	72.1%	78.9%	54.4%	69.7%	76.8%	52.4%	68.2%	71.7%
	T	OpenMEL	61.9%	71.8%	74.6%	57.8%	66.5%	68.2%	63.0%	73.3%	75.6%
	T+V	OpenMEL (w/o HITL)	69.8%	81.0%	83.4%	65.5%	77.4%	80.4%	67.1%	82.2%	85.2%
	T+V	OpenMEL	**69.8%**	**81.0%**	**83.4%**	**65.6%**	**77.4%**	**80.4%**	**67.1%**	**82.2%**	**85.2%**
	T+V	OpenMEL (GPT-4o)	75.1%	84.3%	85.9%	72.6%	81.1%	83.2%	72.4%	87.2%	90.3%

Log Anomaly Identification [ICDE'26]: MaidLog

1. Problem / Limitations

Logs are crucial for alerting on system anomalies, yet existing identification methods do not perform well in resource-constrained scenarios.

Data resource constraints:

- Evolving systems → Generalizable to **OoD data**
- Label-free entries → Applicable **without labels**

Computational resource

- No GPU **for inference** → Be as **lightweight** as possible

2. Key Insight

I can do zero-shot generalization.

I can work on low performance devices.

LLM s ⟷ Decouple ⟷ **Small NNs**

LLMs to generate labels for training;

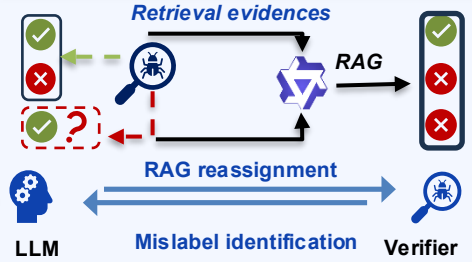
Trained small NNs to deploy for inference.

1. Existing Log Data

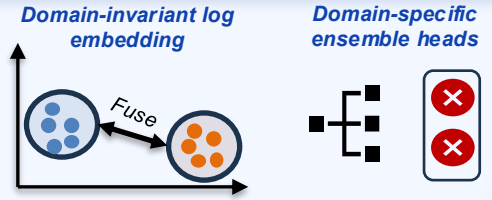
System init at 0x7e
System failed at 0x8b
System reinit at 0x9a
...

w/o label
Multiple domains

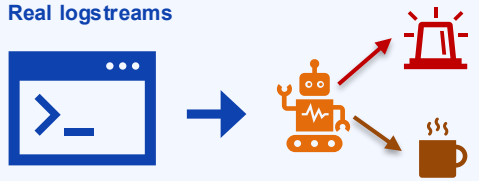
2. LLM-assisted pseudo-labels generation



3. Generalizable detector



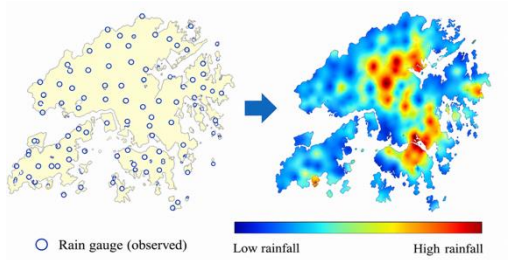
4. Deployment



Efficient Zero-shot and Label-free Log Anomaly Detection for Resource-constrained Systems, ICDE 2026

Data Augmentation [SIGMOD'23]: SSIN

1. Problem



- Reliable high temporal resolution rainfall information is essential for disaster warning.
- Rainfall spatial interpolation estimates rainfall values for locations without stable observations.

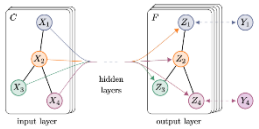
Limitations of Existing Approaches

Traditional methods

$$\hat{z}(v_0) = \sum_{i=1}^{n} \lambda_i * z(v_i)$$

Deterministic functions / statistical assumptions

GNN-based solutions



pre-defined adjacency matrices

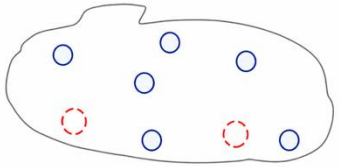
2. Key Insight: Cloze Task

Cloze task with blanks in a sentence

[?] dog [?] so cute.

My dog is so cute.

Spatial interpolation with missing values in a spatial map

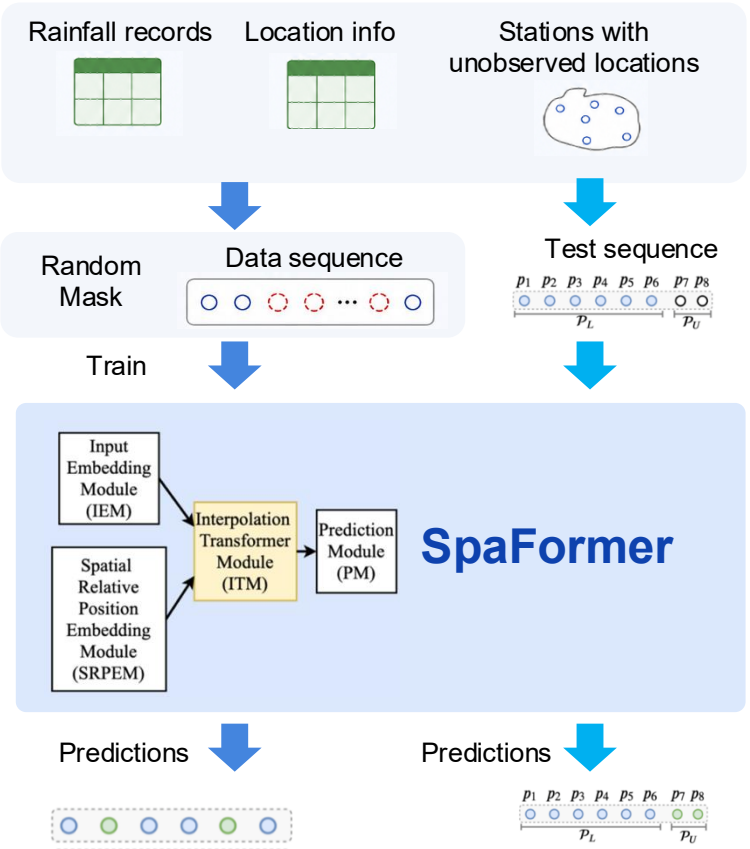


Spatial interpolation is a fill-in-the-blank problem in the spatial domain, similar to the Cloze task.

Self-supervised learning (SSL), widely used in cloze-style tasks, provides a natural solution.

3. Method: SSIN Framework

SSIN Framework



SSIN: Self-Supervised Learning for Rainfall Spatial Interpolation, SIGMOD 2023

Data Cleaning [ICDE'24]: Triple-d

1. Problem / Limitations

Distant supervision produces large training data. But distant supervision introduces noise. High-quality training data is essential.

- **At-least-one assumption** - cannot identify the true relation instance.
- **High-confidence selection** - fails when both prediction and distant supervision are wrong.
- **Dependency paths** - may miss relation semantics.

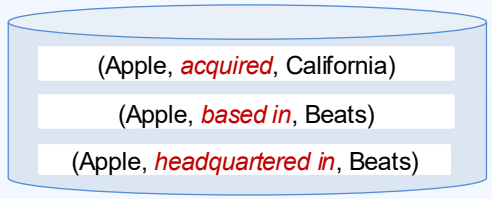
2. Key Insight

Jointly optimize context-preserving pattern replacement and separate noisy instances with non-parametric semantic clustering.

Example: *Apple acquired Beats in California* ⟷ Lack of Ground truth

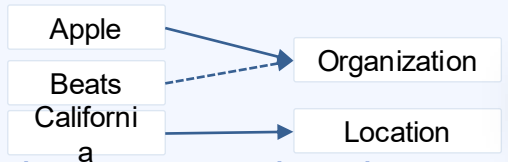
1. Noisy Distant Supervision

Noisy training instances (bag of triples)



2. Module 1: Pattern Replacement

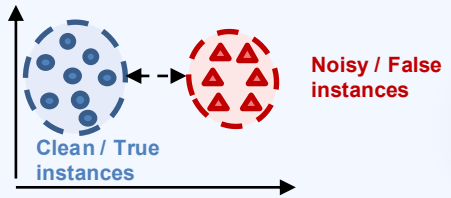
context-preserving pattern replacement via bipartite graph selection



select context-preserving replacements

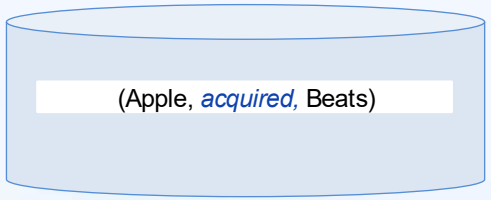
3. Module 2: Semantic Clustering

scalable non-parametric denoising



4. Clean Training Data

Denoised training instances (clean triples)



Triple-d: High-quality Training Data Creation through Denoising Distant Supervision, ICDE 2024

Data Cleaning [VLDB'20]: KBPearl

1. Problem

KBs are crucial for question-answering, recommendations, and RAG,but existing KBs are incomplete and rebuilding a KB is costly.

Canonicalization problem:

Too many synonymous relations or ambiguous entities

J.W. Lennon | He

John Lennon

Linking problem:

Separately processing each triple ignores semantic concepts.

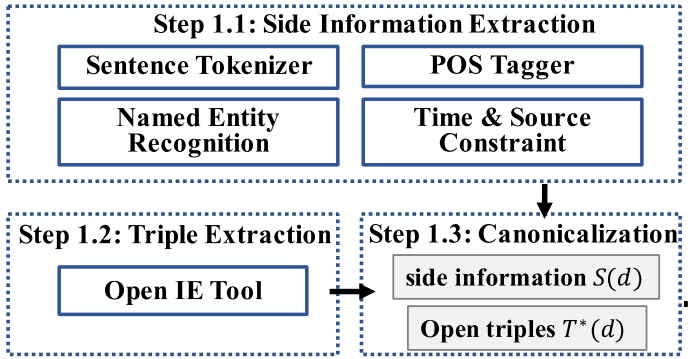
Micheal Jordan | AI

✗ Basketball player ✗ Movie

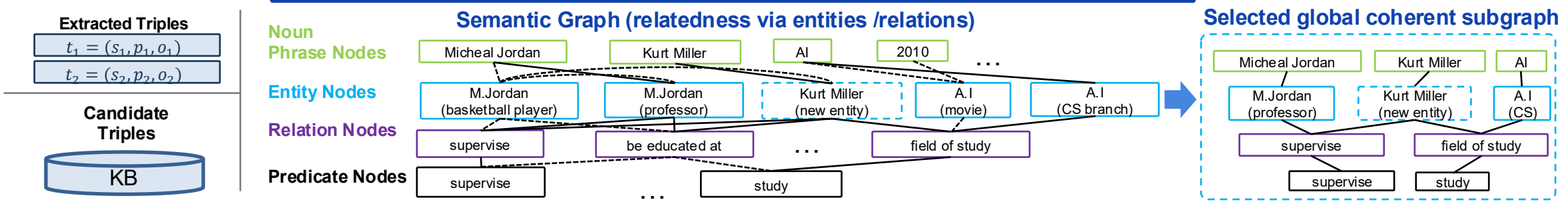
✓ Scientist ✓ Academic field

2. Key Insight

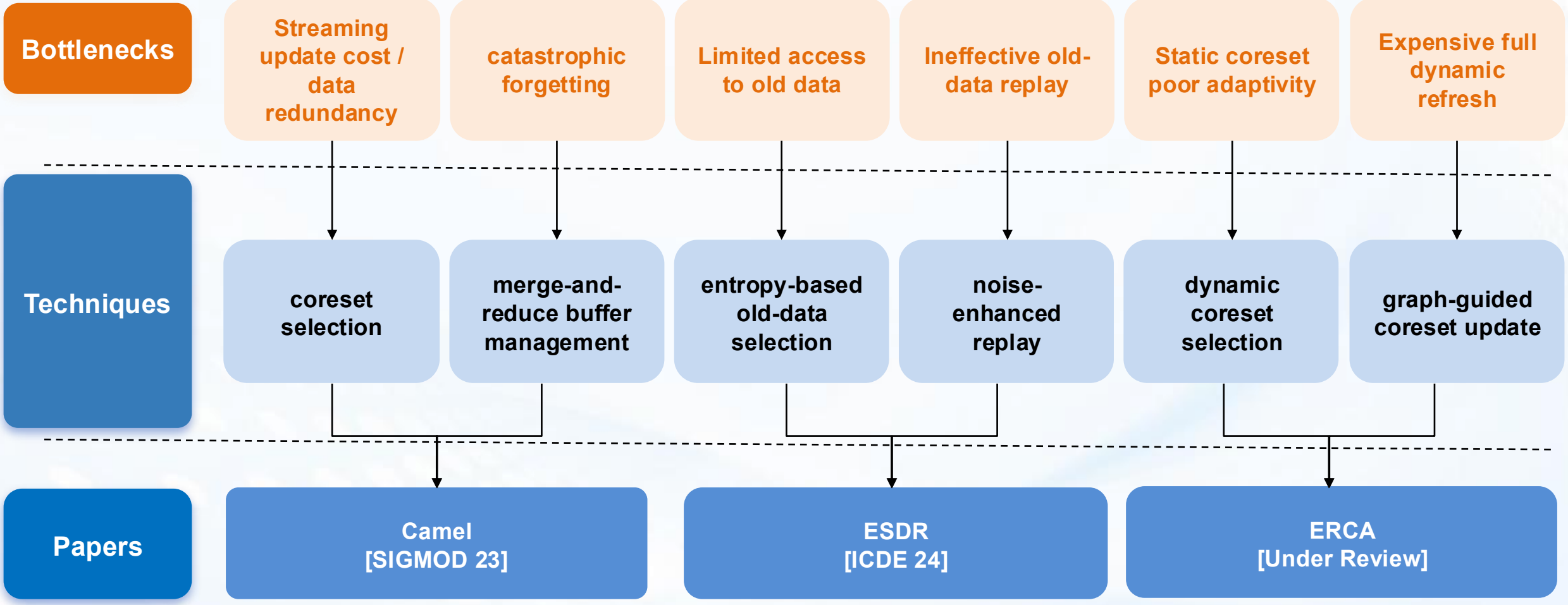
- Canonicalization with **contextual side information**.
- Find a subgraph between extracted triples and candidate triples to ensure **global-coherence.**



3. Method: KBPearl Framework



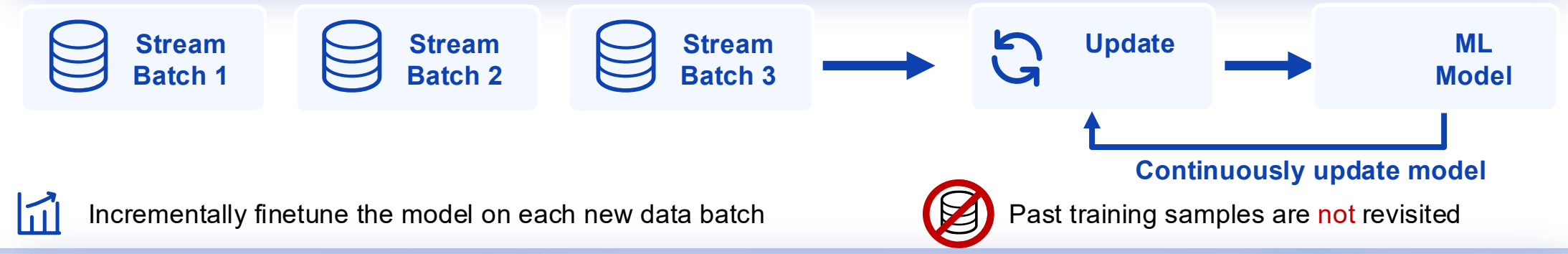
Model Training



Coreset Selection [SIGMOD'23]: Background

Online models must learn from continuous data streams.

A stream learning framework



Incrementally finetune the model on each new data batch

Past training samples are not revisited

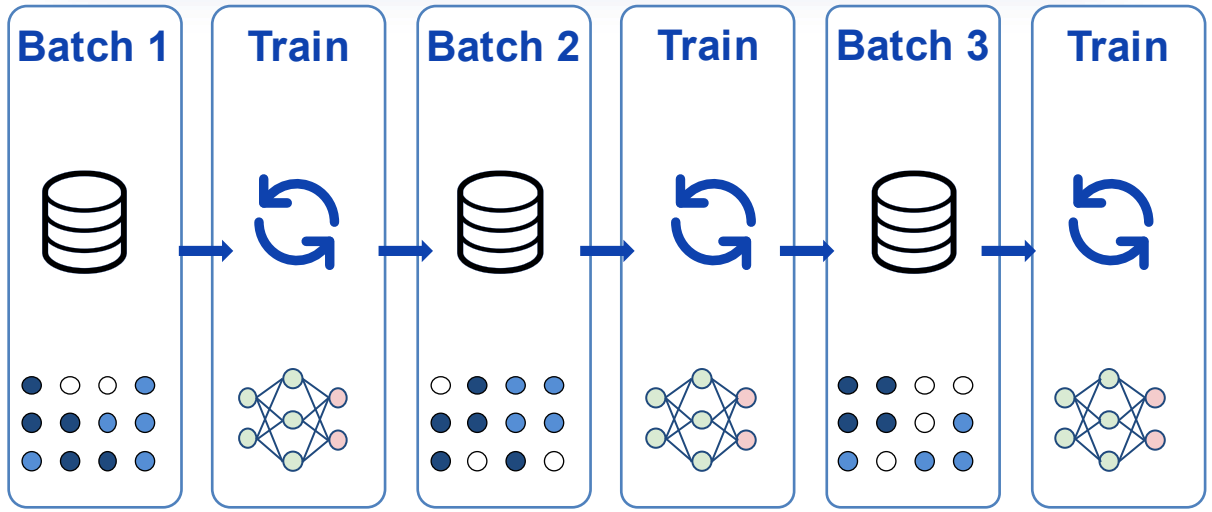
Example Applications

A robot learns new skills and adapts to new situations

A spam filter learns to classify new spams

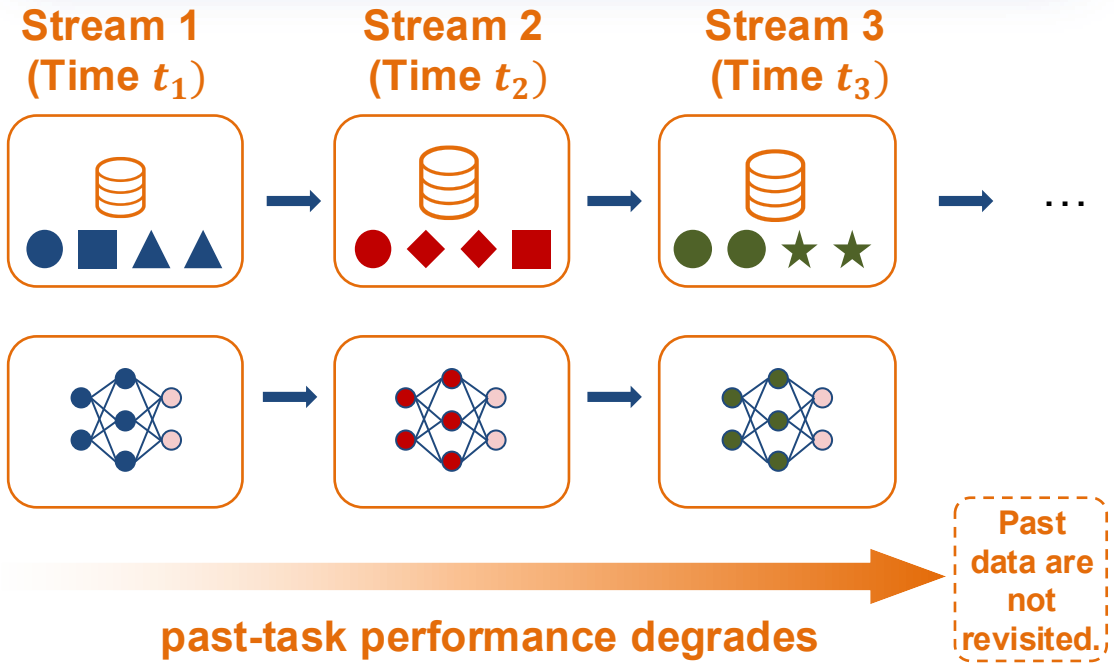
Coreset Selection [SIGMOD'23]: Limitations

Redundant full-batch updates



- Naive stream learning treats every arriving sample as equally useful.
- leading to high update cost and low data efficiency.

Forgetting under non-IID streams

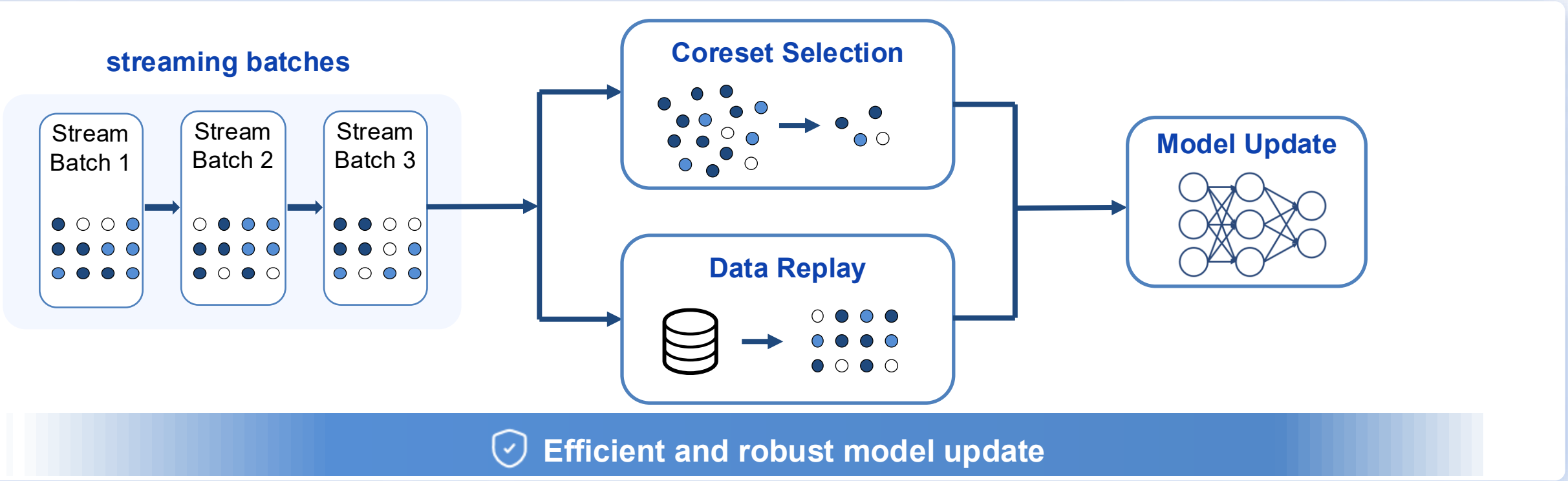


- Without a compact and representative memory, the model gradually loses performance on past distributions.

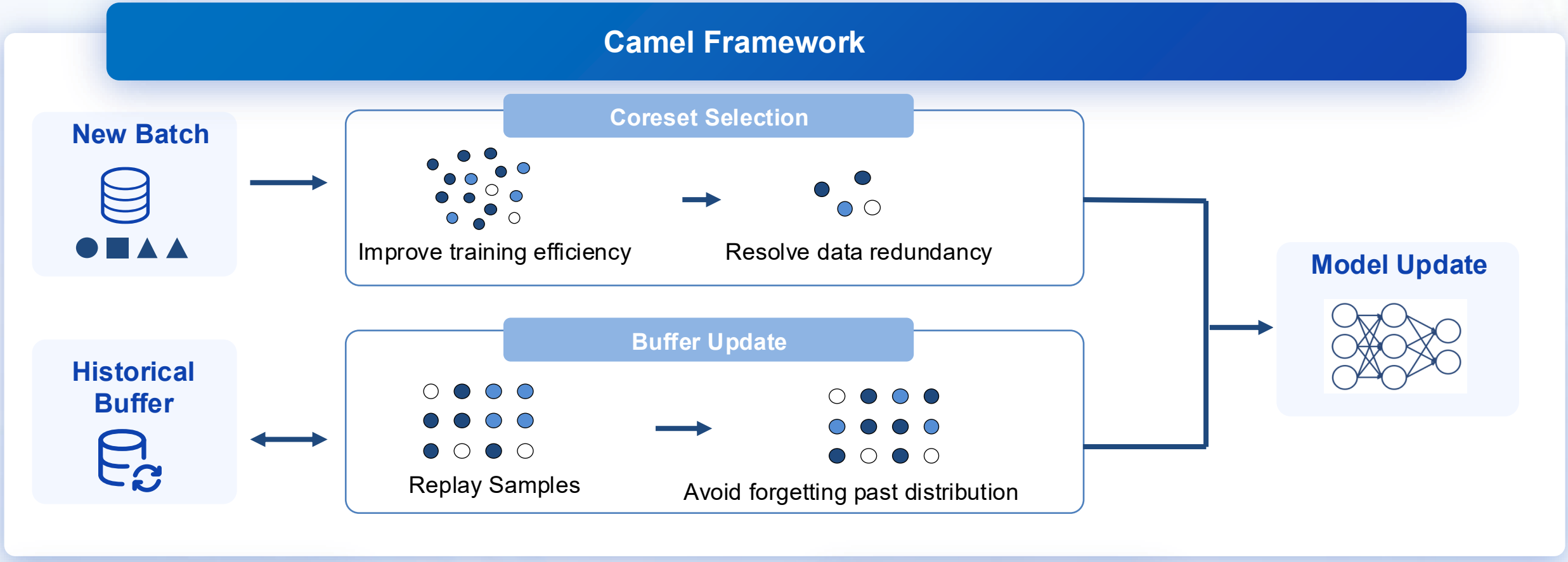
Naïve stream learning is inefficient at large data and vulnerable to forgetting.

Coreset Selection [SIGMOD'23]: Key Insight

Treat both incoming-batch training and historical-memory maintenance as coreset selection problems: select a weighted coreset for efficient one-pass updates and maintain a compact replay buffer through merge-and-reduce.



Coreset Selection [SIGMOD'23]: Framework Overview



Coreset Selection

Select a compact, representative subset from new data to improve training efficiency and reduce redundancy

Buffer Update

replay samples from historical memory to preserve past data distribution and prevent forgetting

Coreset Selection [SIGMOD'23]: Detailed Modules

Coreset selection approximates full-batch training with a compact weighted subset; buffer management preserves historical memory through merge-and-reduce.

Module 1: Coreset Selection

Camel selects a small weighted coreset from each incoming batch for one-pass model update.
The goal is to make training on the coreset approximate training on the full batch under a worst-case parameter view.

$$w* \in \operatorname{argmin}_{w \in \mathbb{R}_+^n, \|w\|_0 = m} \max_{\theta \in \Theta} L(B, w, \theta),$$

Approximation error

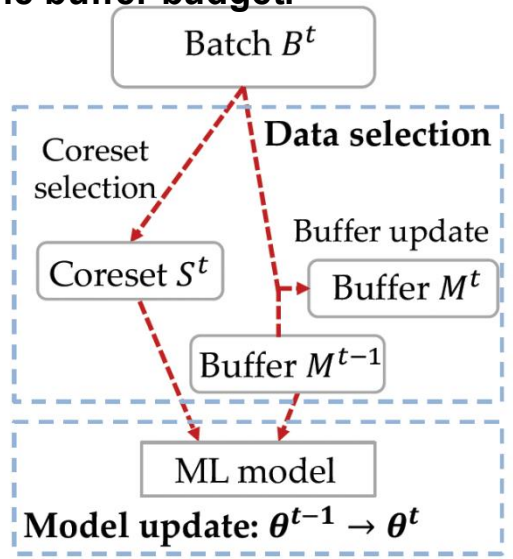
$$L(B, w, \theta) \triangleq \frac{1}{n} \left| \sum_{i=1}^{n} l(x_i, y_i; \theta) - \sum_{j=1}^{n} w_j l(x_j, y_j; \theta) \right|$$

Loss values on the full set B

Loss values on the coreset S

Module 2: Buffer Management

Camel treats buffer maintenance as another coreset problem: merge old memory with the new batch, then reduce back to the buffer budget.



Coreset Selection [SIGMOD'23]: Experimental Performance

Camel addresses both streaming update cost and catastrophic forgetting by jointly selecting representative new data and maintaining a compact historical memory.

Experiment 1: Coreset selection

Coreset outperforms random sampling and even the full batch algorithm in model accuracy on a variety of datasets.

Table 2: Average test accuracy (%) of Coreset and Random varying sample ratio r. Sample ratio 100% corresponds to the Full-Batch algorithm. Best results are marked in bold, and the results better than Full-Batch are underlined.

Data Stream	Method	Sample Ratio r 30%	40%	50%	60%	70%	80%	90%	100%
CUB200	Random	47.91 ± 0.45	48.65 ± 0.32	49.1 ± 0.62	49.84 ± 0.22	50.14 ± 0.13	50.51 ± 0.18	50.55 ± 0.25	50.76 ± 0.37
	Coreset	**48.78 ± 0.22**	**49.57 ± 0.23**	**50.12 ± 0.36**	**50.13 ± 0.25**	**50.61 ± 0.14**	**50.74 ± 0.21**	**50.66 ± 0.17**	–
CIFAR10	Random	75.58 ± 0.59	76.81 ± 0.85	77.81 ± 1.02	79.55 ± 0.64	80.07 ± 0.43	80.57 ± 0.56	80.84 ± 0.83	81.25 ± 0.54
	Coreset	**76.63 ± 0.48**	**77.7 ± 0.67**	**79.03 ± 0.75**	**79.96 ± 0.5**	**80.63 ± 0.46**	**80.82 ± 0.42**	**81.53 ± 0.32**	–
MNIST	Random	95.03 ± 1.27	95.66 ± 0.38	96.95 ± 0.51	97.47 ± 0.6	97.71 ± 0.6	97.8 ± 0.42	98.2± 0.11	97.99 ± 0.27
	Coreset	**95.99 ± 0.65**	**97.26 ± 1.25**	**97.31 ± 0.55**	**97.58 ± 0.5**	**97.9 ± 0.36**	**98.07 ± 0.34**	**98.27 ± 0.23**	–
SVHN	Random	88.51 ± 0.38	89.14 ± 0.21	90.17 ± 0.2	90.82 ± 0.39	91.13 ± 0.41	90.97 ± 1.58	91.88 ± 0.28	92.2 ± 0.17
	Coreset	**89.56 ± 0.32**	**89.85 ± 0.45**	**90.79 ± 0.42**	**91.16 ± 0.3**	**91.64 ± 0.38**	**91.82 ± 0.23**	**92.2 ± 0.38**	–
CORe50	Random	69.29 ± 0.43	68.99 ± 0.27	69.21 ± 0.59	68.61 ± 0.55	69.06 ± 0.42	68.68 ± 0.34	68.54 ± 0.4	69.36 ± 0.31
	Coreset	**69.45 ± 0.45**	**69.66 ± 0.59**	**69.58 ± 0.27**	**69.45 ± 0.47**	**69.59 ± 0.56**	**69.46 ± 0.56**	**68.97 ± 0.52**	–
Covtype	Random	55.53 ± 0.37	56.59 ± 0.56	57.31 ± 0.8	57.37 ± 0.33	58.0 ± 0.34	58.2 ± 0.31	58.44 ± 0.24	58.49 ± 0.23
	Coreset	**56.12 ± 0.79**	**56.86 ± 0.79**	**57.53 ± 0.4**	**57.84 ± 0.69**	**58.38 ± 0.29**	**58.44 ± 0.26**	**58.91 ± 0.21**	–
KDD99	Random	94.01 ± 0.27	94.15 ± 0.13	94.22 ± 0.26	94.3 ± 0.28	94.28 ± 0.27	94.16 ± 0.26	94.27 ± 0.38	94.39 ± 0.33
	Coreset	**94.13 ± 0.37**	**94.20 ± 0.21**	**94.23 ± 0.43**	**94.33 ± 0.19**	**94.45 ± 0.13**	**94.46 ± 0.35**	**94.47 ± 0.26**	–
Full SVHN	Random	94.43 ± 0.2	94.72 ± 0.13	95.05 ± 0.1	95.15 ± 0.3	95.41 ± 0.13	95.57 ± 0.16	95.68 ± 0.08	95.78 ± 0.08
	Coreset	**94.58 ± 0.12**	**95.06 ± 0.08**	**95.35 ± 0.07**	**95.62 ± 0.07**	**95.69 ± 0.18**	**95.76 ± 0.14**	**95.8 ± 0.19**	–
Clothing1M	Random	46.04 ± 0.15	46.43 ± 0.24	47.09 ± 0.28	47.42 ± 0.37	**47.76± 0.2**	47.9 ± 0.15	48.07 ± 0.52	48.12 ± 0.17
	Coreset	**46.91 ± 0.19**	**47.18 ± 0.33**	**47.44 ± 0.39**	**47.55 ± 0.24**	47.74 ± 0.31	**47.98 ± 0.17**	**48.31 ± 0.35**	–
ImageNet	Random	48.48 ± 0.12	48.56 ± 0.09	48.74 ± 0.06	48.93 ± 0.09	**49.03± 0.05**	49.05 ± 0.05	49.17 ± 0.08	49.28 ± 0.08
	Coreset	**48.52 ± 0.08**	**48.66 ± 0.04**	**48.77 ± 0.04**	**48.98 ± 0.08**	49.01 ± 0.1	**49.13 ± 0.06**	**49.24 ± 0.06**	–

Experiment 2: Buffer Management

Merge-and-Reduce (MR) algorithm can better avoid catastrophic forgetting while being more efficient.

Data Stream	Distribution	Metric	Data Replay					Distillation	
			MR	**Agglo**	Random	GSS	MIR	iCaRL	DER
split-CIFAR10	Class-incremental	Accuracy	47.13	**48.57**	44.69	43.48	18.24	43.32	33.78
		Runtime	247.41	240.32	243.68	1122.41	367.15	431.27	189.56
split-CIFAR100	Class-incremental	Accuracy	**50.23**	50.13	49.55	40.85	43.3	8.75	36.74
		Runtime	246.86	240.73	244.21	8327.78	386.91	723.15	193.82
split-SVHN	Class-incremental	Accuracy	82.12	**82.79**	81.26	58.87	73.59	68.92	79.39
		Runtime	343.96	339.17	341.25	2093.47	459.06	542.51	269.42
CORe50	Domain-incremental	Accuracy	48.70	**50.25**	46.27	42.52	48.82	-	46.09
		Runtime	529.24	537.09	522.27	2688.85	839.01	-	468.71
CORe50-iid	IID	Accuracy	**55.85**	54.46	48.0	50.91	52.68	-	50.1
		Runtime	538.17	544.45	535.36	2130.85	837.91	-	454.28
CIFAR10	IID	Accuracy	64.79	**65.37**	63.46	62.02	52.99	-	53.67
		Runtime	208.75	209.25	213.70	960.20	399.14	-	190.86

Data Replay [ICDE'24]: ESDR

Background

Continuous unlabeled data arrives over time, making one-shot training insufficient

Limited access to old data

Old data is only partially accessible due to privacy or efficiency issues.

Catastrophic forgetting

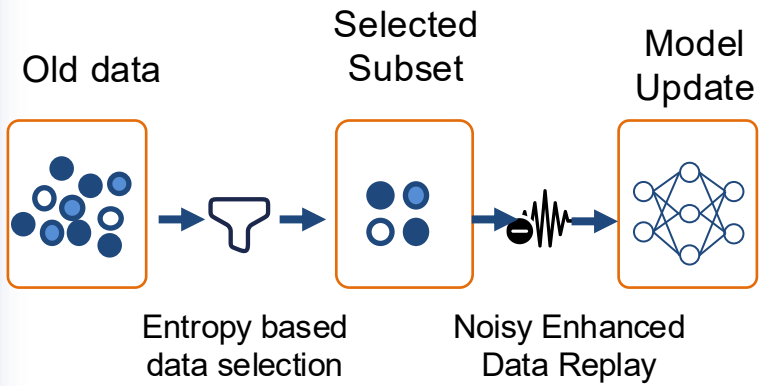
The model must be updated over time without losing old knowledge.

Ineffective old-data replay

Existing UCL methods lack principled sample selection and replay.

Key Insight

Improve old-data utilization through **entropy-based representative sample selection and noise-enhanced replay** of past knowledge.

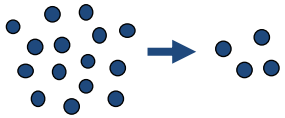


ESDR combines representative sample selection and noise-enhanced replay to mitigate forgetting under limited old-data access.

Method

Module 1

Entropy-based Data Selection

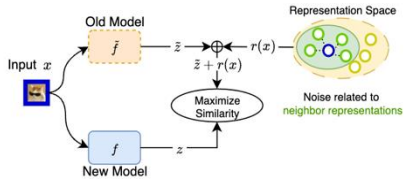


$$M^* = \arg\max_{M \subseteq X} Entropy\,(M)$$

Select representative old samples that preserve maximal information of the full old dataset.

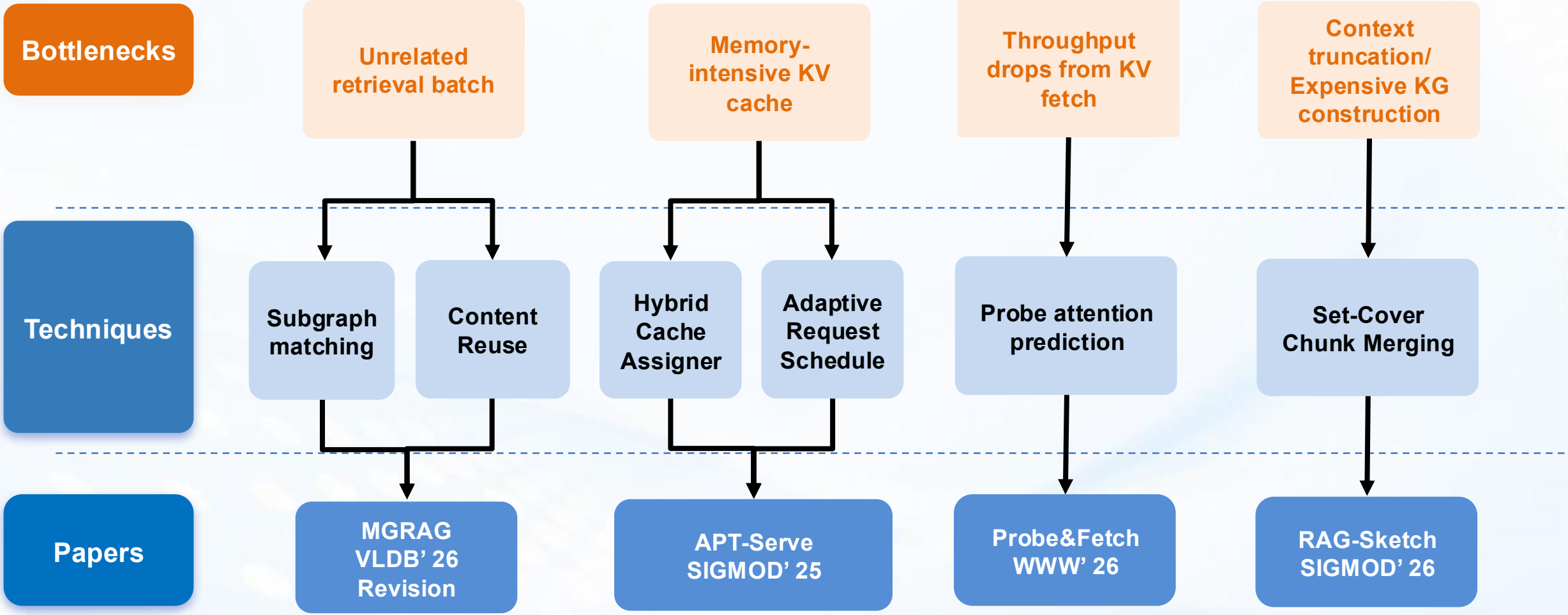
Module 2

Noise-enhanced Data Replay



Distill knowledge from the old model with representation-level noise to cover neighboring samples more effectively.

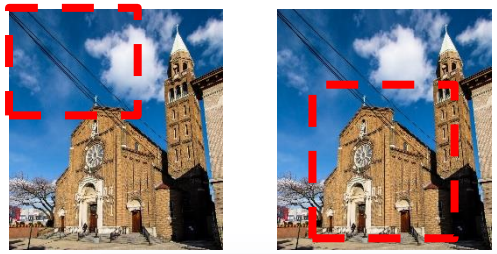
Cache and Batch Technique



Adaptive Batch [VLDB'26 R]: MGRAG

1. Problem/Limitations

Different queries focus on different visual contents on same image



- ✗ **Query-unaware** indexing
- ✗ Fine-grained RAG: **High latency**
- ✗ Query-aware infeasible due to **corpus size and query number overhead**

2. Key Insights

- Construct query specific multimodal KGs only with the most query similar contents
- Formulate the KG retrieval as path-aware semantic subgraph matching problem

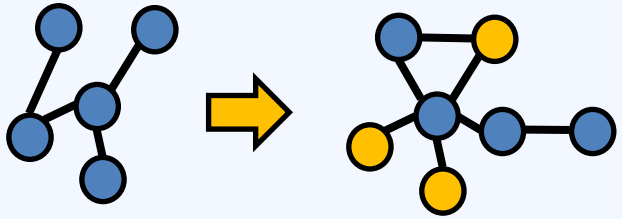
✓ Query-aware ✓ Adaptiveness ✓ Feasible latency

3. Method: Query-aware Multimodal Retrieval

Q: What's the weather in the image?

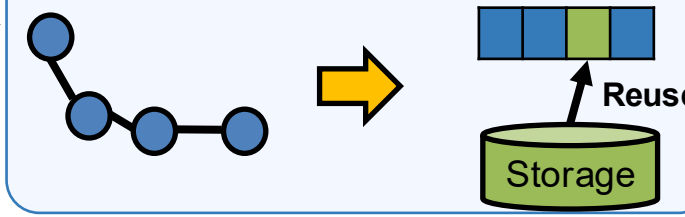
Query-aware incremental KG construction

Incrementally update a query specific KG, until find a matched subgraph to the query graph



Topological-aware KV cache

Reuse caches based on the retrieved subgraph's topological feature and tokens' cross attention



Subgraph Augmented Inference

MGRAG: A Graph-based Multimodal Retrieval-augmented Generation System with Large Language Models, VLDB 26 Revision

KV Cache [SIGMOD'25]: Background

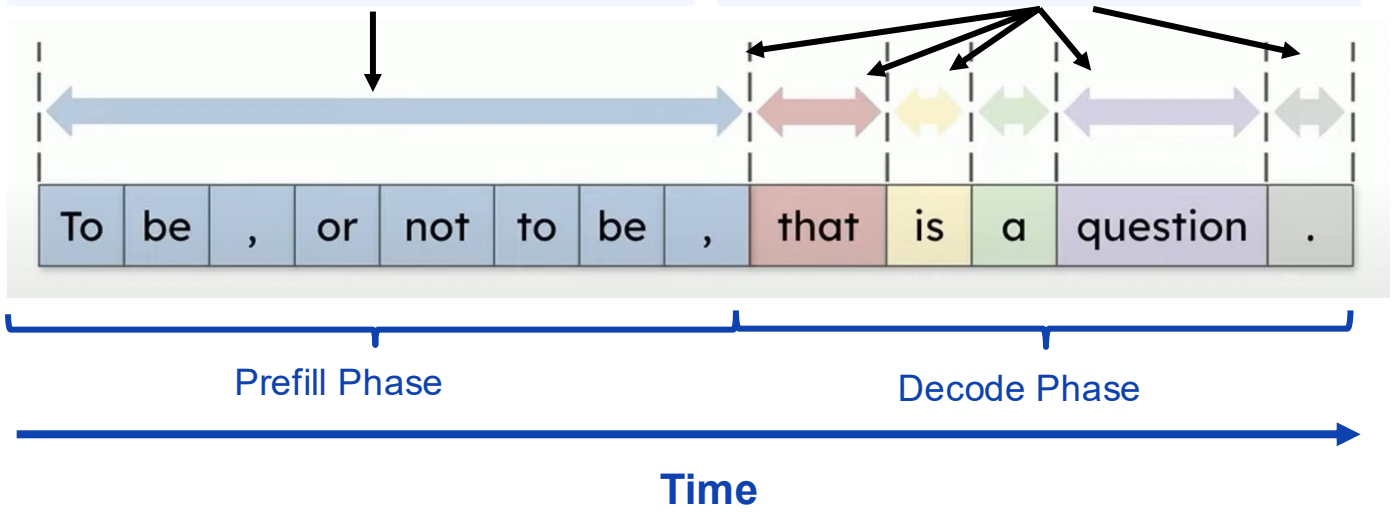
LLM Inference Breakdown

Prefill Phase Process the entire input.

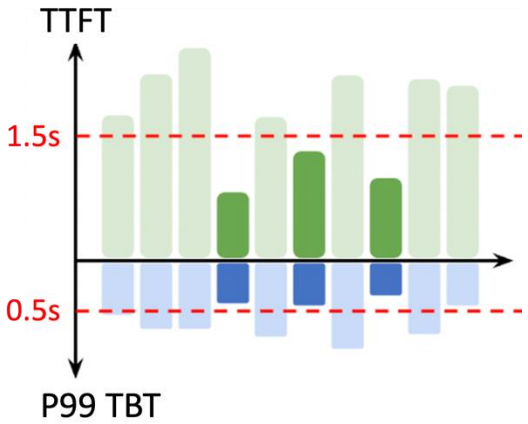
Decode Phase Generate output tokens one-by-one.

Time to First Token (TTFT)
Latency to generate the first token

Time between Token (TBT)
Latency to generate each subsequent token



Both TTFT and TBT are critical to user experience!

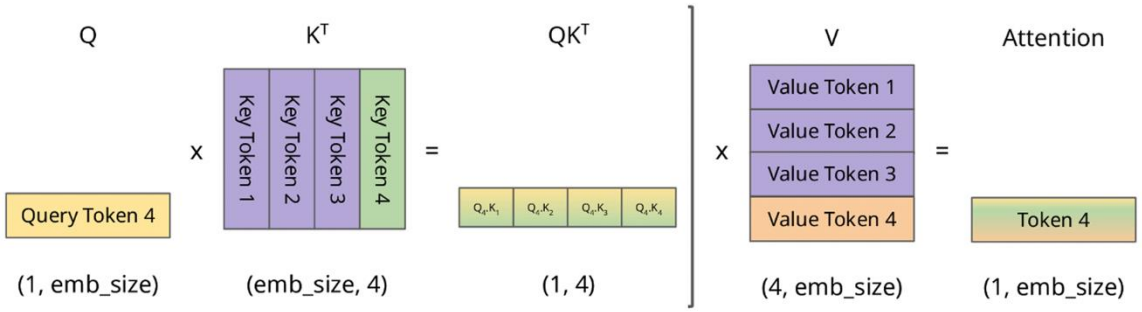


Source: "DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving, OSDI 24".
Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving. SIGMOD'25

KV Cache [SIGMOD'25]: Limitations

Issue I: Limited Batch Size

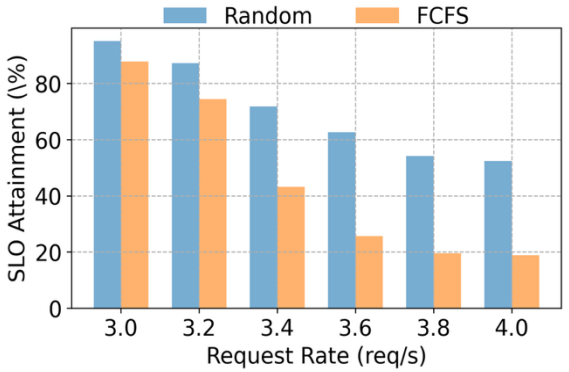
- KV cache: reusable key and value for past tokens
- Overwhelming KV cache
- Limited GPU memory

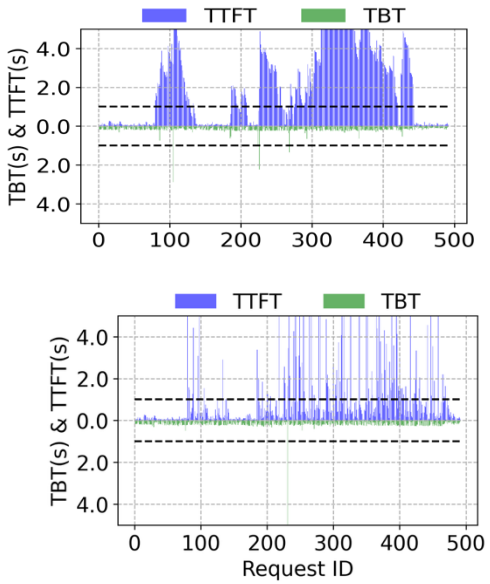


: KV Cached Tokens : New Tokens

Issue II: Suboptimal Batch Composition

First-Come-First-Serve (FCFS) scheduling: Performs worse than random scheduling 🤔





First-Come-First-Serve (FCFS) v.s. Random

Source: “Transformer KV Caching Explained - Medium”.

KV Cache [SIGMOD'25]: Key Insight

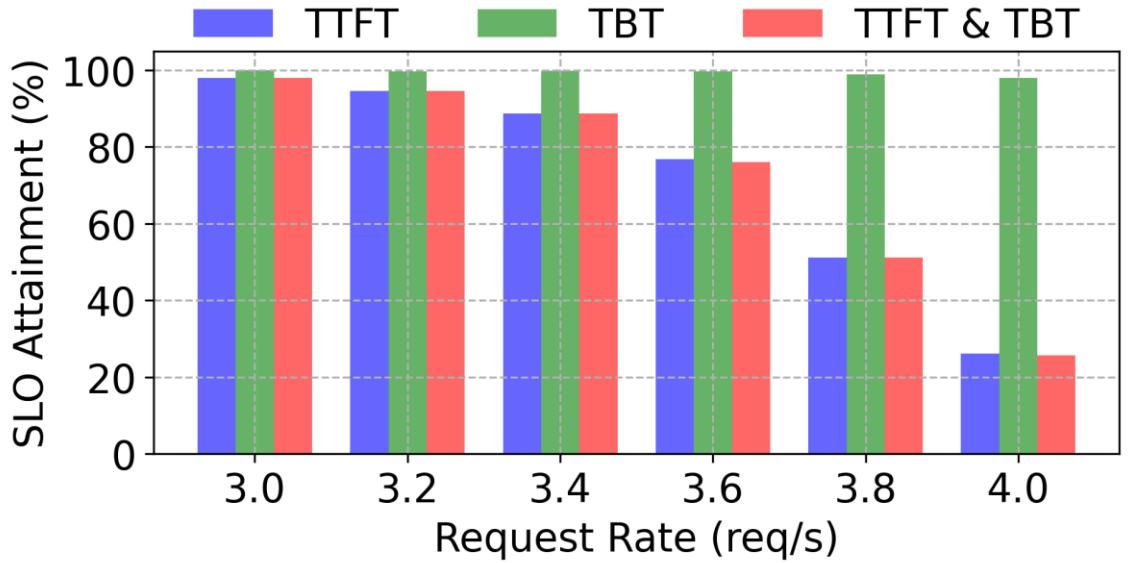
Bottlenecks

Time To First Token (TTFT) is the main bottleneck overall

- Extensive use of **memory-intensive KV cache** prevents enlarged batch size.
- Default FCFS scheduling policy enforces **rigid batch composition.**

Insights

- Use **hybrid KV cache and hidden cache scheme** to reduce memory overhead
- Employ an **adaptive runtime scheduling mechanism** to schedule requests efficiently



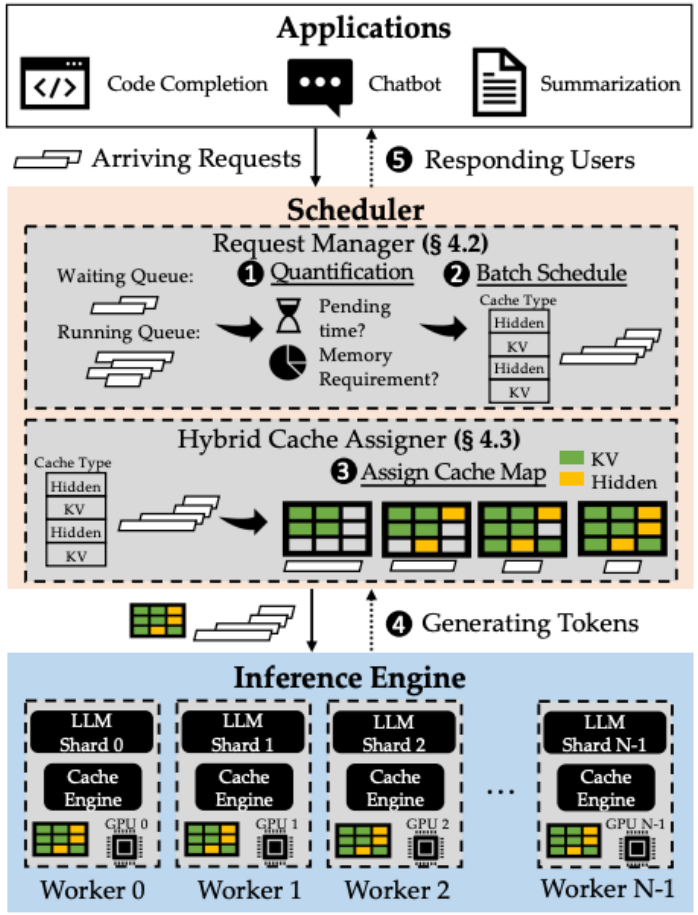
KV Cache [SIGMOD'25]: Framework Overview

Adaptive Request Scheduling

Request Manager determines the optimal batch schedule at the start of each inference .

Hybrid Cache Utilization

Hybrid Cache Assigner seamlessly coordinates the unified usage of both **memory-intensive** KV cache and **memory-efficient** hidden cache.

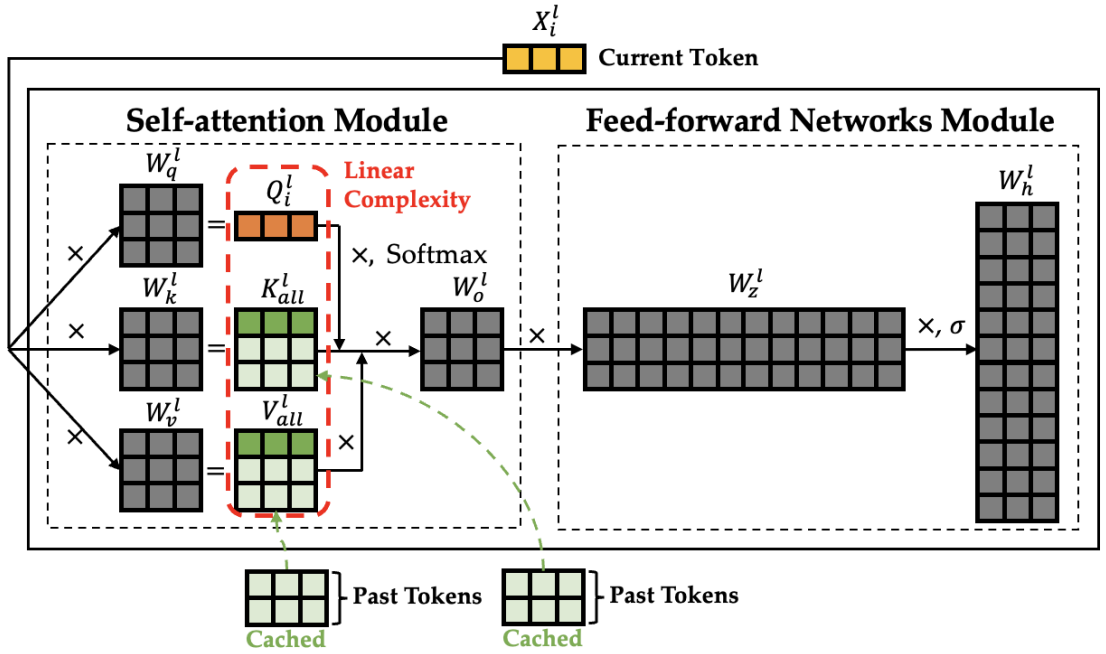


KV Cache [SIGMOD'25]: Module 1

KV cache: trade storage for time

Smaller batch size

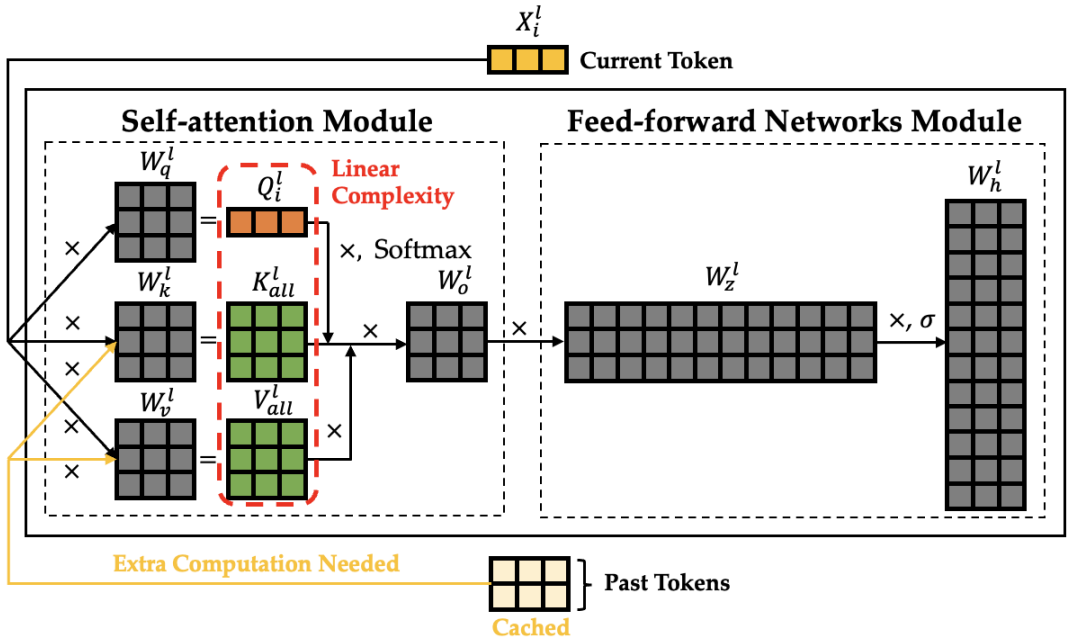
Higher computation speed



Hidden cache: trade time for storage

Larger batch size

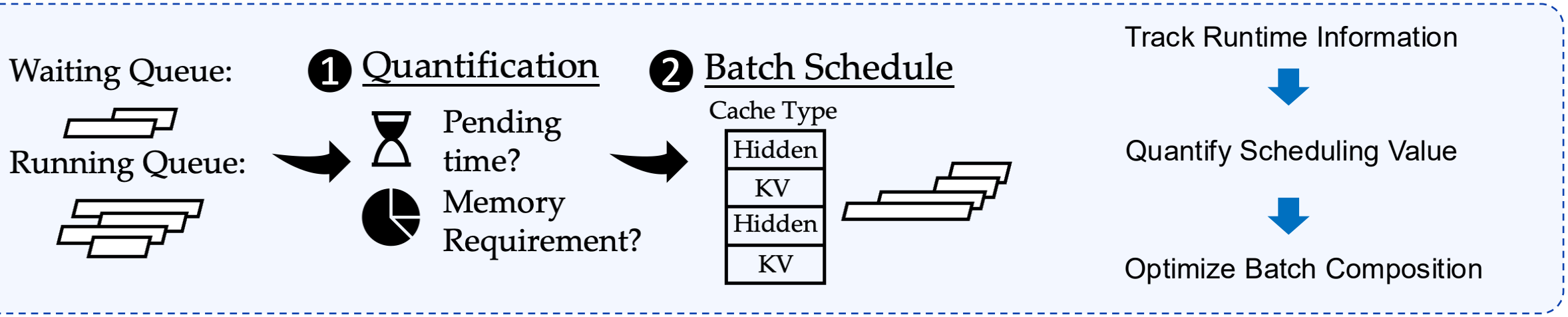
Lower computation speed



KV Cache [SIGMOD'25]: Module 2

Two online decisions

- **Which requests** to be scheduled
- **Which type of** cache for a given request



Scheduling Value $v_i^e = p_i^e - \beta_i^e(|W^e| + |R^e|)t_i^e$ (request i, target iteration e)

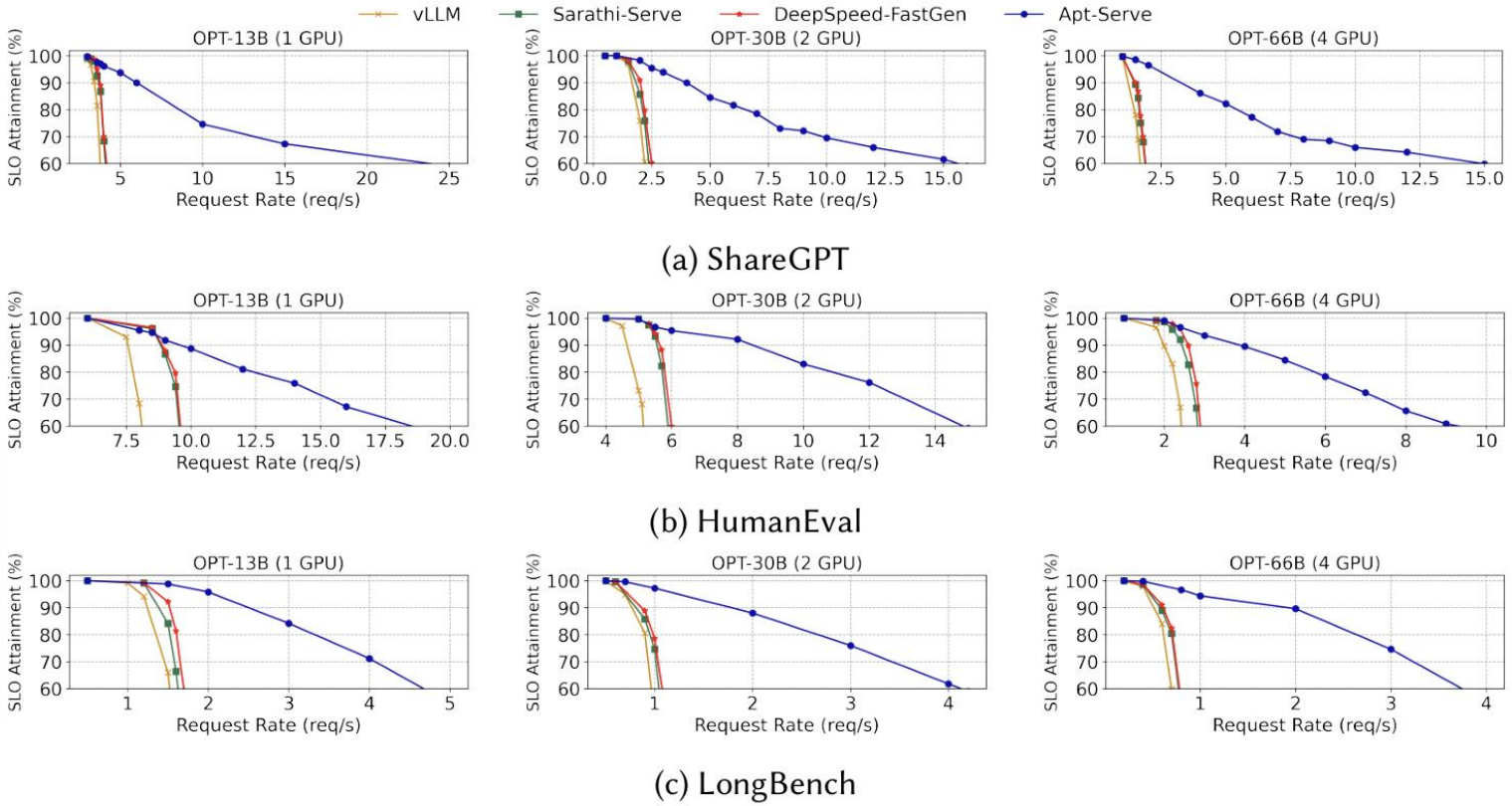
- p_i^e: **Intuition I: positively related to the pending time**
- $\beta_i^e(|W^e| + |R^e|)t_i^e$: **Intuition II: penalty term for hidden cache usage**

KV Cache [SIGMOD'25]: Experimental Performance

Chatbot Application: SharedGPT | **Code Completion: HumanEval** | **Summarization: LongBench**

Up to **8.8x** Higher effective throughput

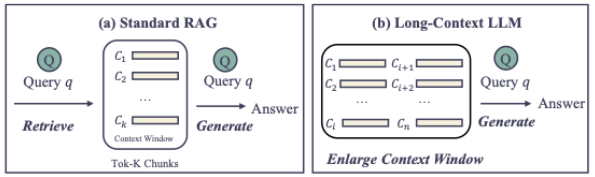
2.0x Higher effective throughput on average



Adaptive Batch [SIGMOD'26]: RAG-Sketch

1.Problem/Limitation

Equipping LLMs with **large external corpus** for Question Answering (QA) becomes critical.



Limitations of Existing Approaches

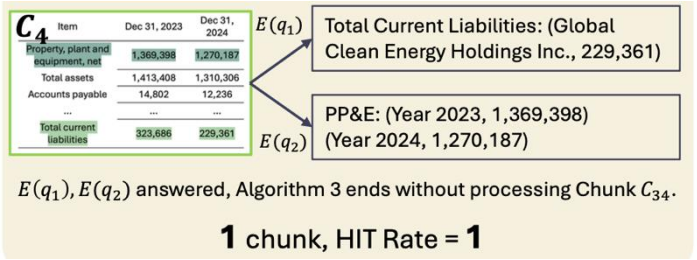
- Limited retrieval (misses scattered evidence)
- Truncation of long contexts
- Expensive KG construction costs

2.Key Insight

- **Decompose** complex questions
- **Improves retrieval quality** through both semantic and presence objectives
- **Minimizes token costs** through **adaptive batching**

Chunk	Presence	Semantic
C_4	1	0.626
C_{32}	0	0.736
C_{34}	0	0.750

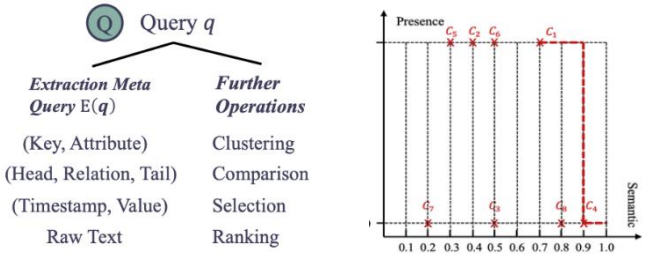
Chunk	Presence	Semantic
C_4	1	0.663
C_{29}	0	0.773
C_{34}	0	0.774



We **adaptively batch queries** that correspond to the same chunk (e.g., C_4 for q_1 and q_2) through set cover chunk merge.

3. Method: RAG-Sketch Framework

- **Query Decomposition Module**: decompose complex queries;
- **Skyline Retrieval Module**: adaptively retrieve required evidence;
- **Set Cover Chunk Merging Module**: we minimize the token costs through batching with theoretical guarantees.



$C_1, T_1 = 300$ $C_2, T_2 = 100$ $C_3, T_3 = 150$

$E(q_1)$	1	1	0
$E(q_2)$	0	1	0
$E(q_3)$	0	1	1
$E(q_4)$	1	0	1
$E(q_5)$	1	0	1

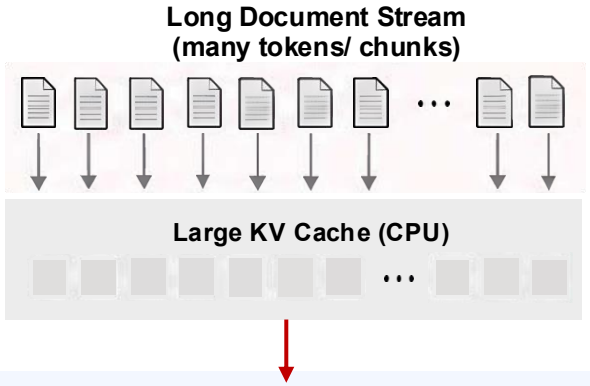
T_i is the token costs

▭ represents the selected chunks

Adaptive Search [WWW'26]: Probe-and-Fetch

1.Problem/Limitation

1. **Web-scale search incurs high memory and latency overhead**
2. **Synchronous KV fetch causes a massive throughput reduction**



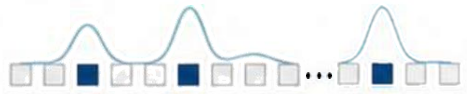
Synchronous Fetch

blocking transfer

~35% throughput drop

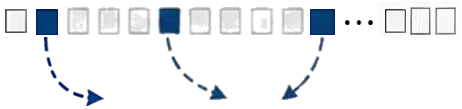
2.Key Insight

1. **Probe attention prediction before verification**



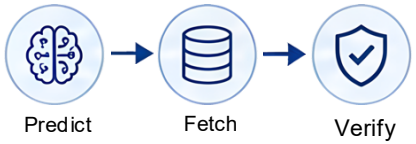
Predict attention over long context and find salient positions.

2. **Asynchronous Fetching**



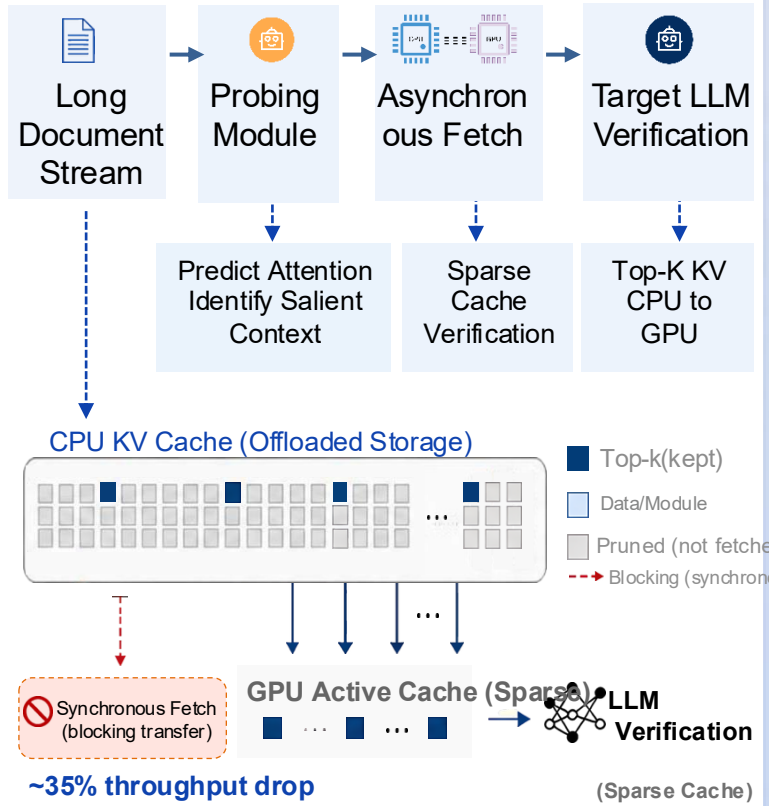
Top-k KV CPU to GPU.

3. **Hide transfer latency by pipelining**



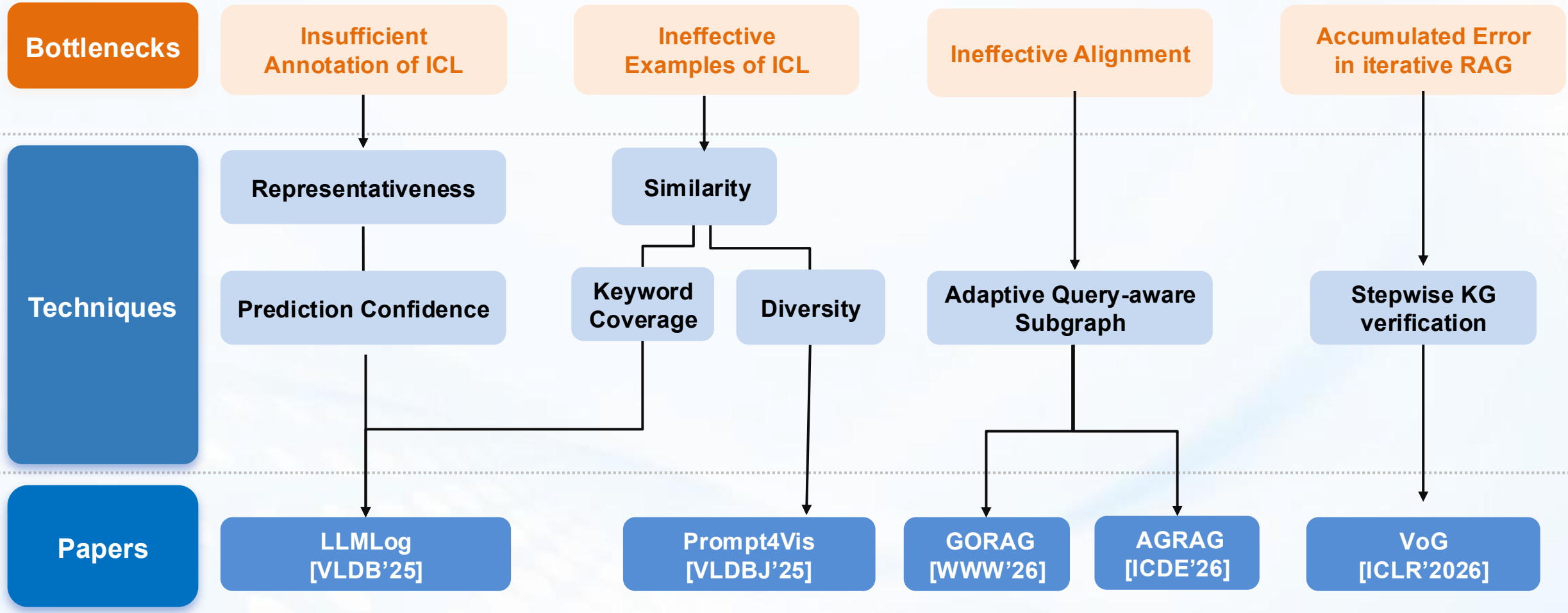
Pipeline context prediction, data movement, and verification.

3.Method/Framework



Li, Yuchen, et al. "Probe-and-Fetch: Dynamic KV Cache Pruning for Accelerated Long-Context Inference in Web-Scale AI Search." WWW 2026

Reliable Generation



In-context Learning [VLDB'25]: Background

System logs are converted into into structured templates for efficient analysis

System Logs

Date	IP	Request	Status	Resource
24-05-10	dsa.ust.hk	GET	500	/submit/form
24-05-11	172.16.1.8	POST	404	/resource/a
24-05-11	cse.ust.hk	GET	503	/service
24-05-12	10.10.2.9	POST	200	/upload/file

In-context Learning

Log Template Generation

Templates

[DATE] [IP] <POST> [STATUS] [RESOURCE]

[DATE] [IP] <POST> [STATUS] [RESOURCE]

[DATE] [IP] <GET> [STATUS] [RESOURCE]

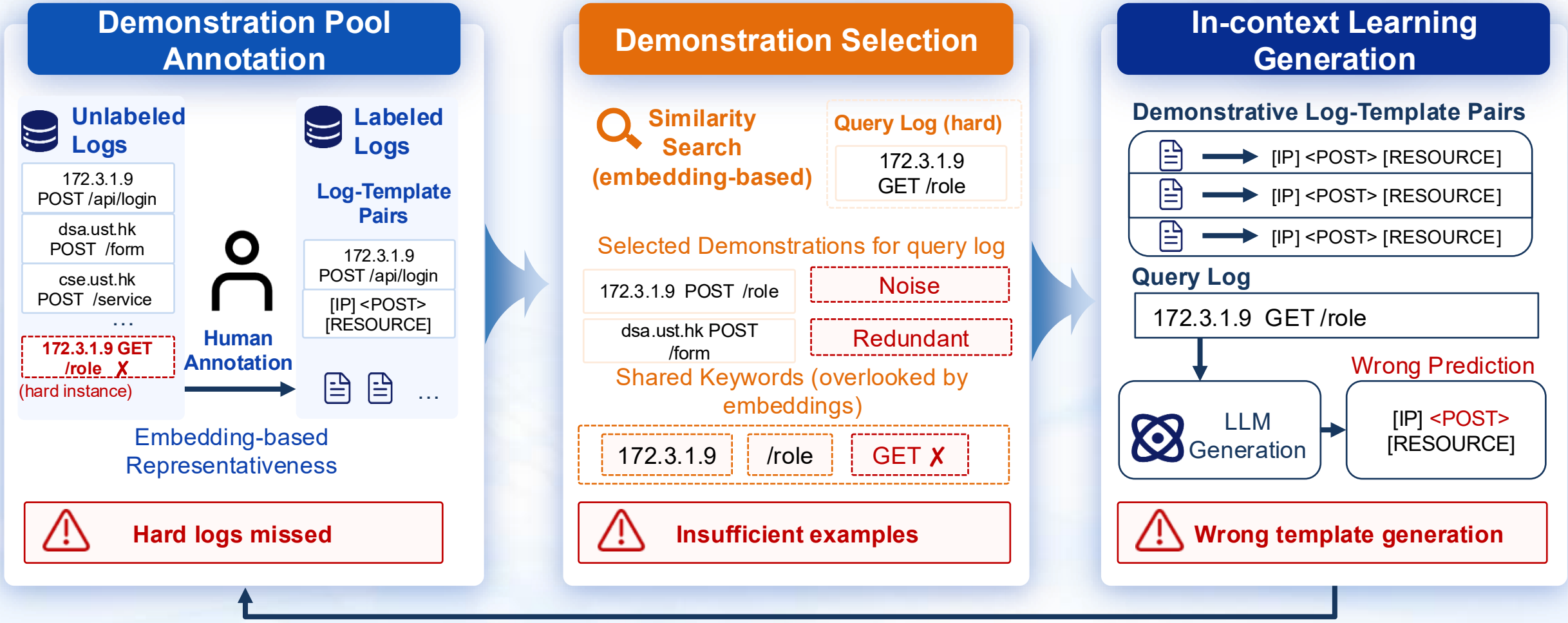
Track error-prone keyword: [STATUS]

Template generation helps maintainers analyze massive logs and identify problematic patterns.

LLMLog: Advanced Log Template Generation via LLM-driven Multi-Round Annotation, VLDB 2025

In-context Learning [VLDB'25]: Limitations

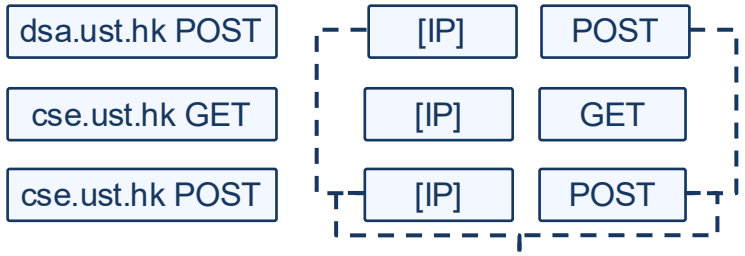
- **Insufficient annotation (missing hard logs).**
- **Ineffective examples (Similarity-based demonstration selection).**



In-context Learning [VLDB'25]: Key Insights

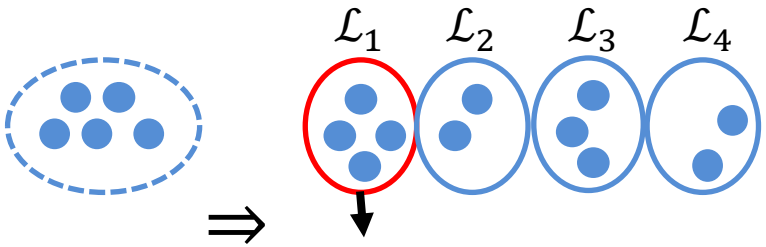
Keyword Coverage Similarity-based Representativeness

Logs that **share more keywords** are more similar.



2 shared keywords
Edit distance = 0

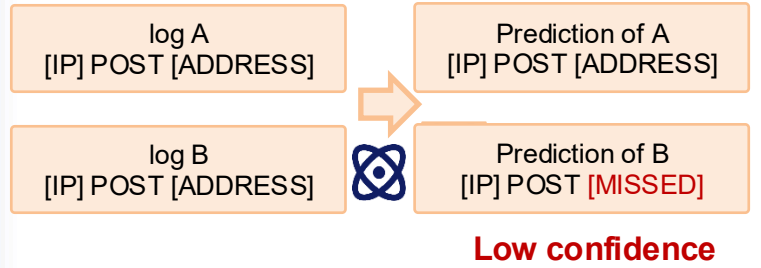
Ideal labeled logs should be **similar to as many** unlabeled logs as possible.



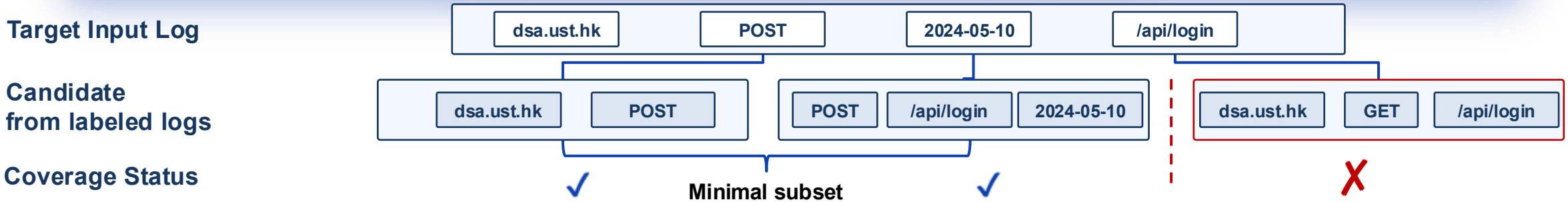
Most representative

Confidence in Annotation

Logs with **low prediction confidence** are worth annotating.



Adaptive Demonstration Selection (Set Cover)



In-context Learning [VLDB'25]: Detailed Modules

Step 1: Demonstration Log Annotation

Input: candidate unlabeled logs, each has representative log set + LLM predicted template.

Output: selected logs for human annotation.

Low-confidence logs (hard instances)

Also prefer logs **similar to many** logs.

Rank	Candidate log	Representative Score	Confidence Score
1	dsa.ust.hk POST /api 500	0.87	**0.31**
2	cse.ust.hk GET /login 404	0.76	**0.21**
3	cse.ust.hk POST /index 200	0.62	0.71
4	cse.ust.hk POST /about 201	0.51	0.63

Selected for annotation

1
2
3

Goal: prioritize representativeness + low confidence.

Solution: Greedy, 1-1/e approx.

Step 2: Adaptive Demonstration Selection

Input: an unlabeled log + labeled log set.

Output: demonstrations for the unlabeled log.

Input Log Keywords

dsa.ust.hk | POST | /api | 500

Labeled Demonstrations	Coverage of input keywords			
	dsa.ust.hk	POST	/api	500
dsa.ust.hk POST /api 500	✓	✓	✓	✓
cse.ust.hk POST /index 200	✓	✓	✗	✓
cse.ust.hk GET /login 404	✓	✗	✗	✓

Goal: minimum demonstrations, full keyword coverage.

Solution: Greedy, 1+ln *n* approx.

In-context Learning [VLDB'25]: Experimental Performance

- In **16 log template generation datasets**, our **LLMLog** is **the most accurate** method.
- **LLMLog** also incurs **less average generation time** and **API cost** than baselines.

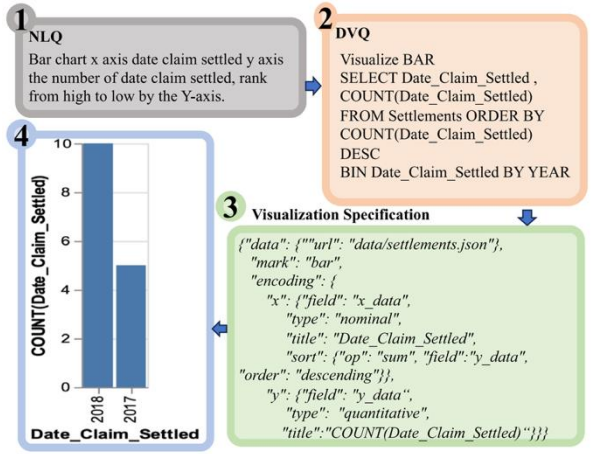
Dataset	Drain			LogPPT			DivLog			AdaICL			LLMLog (Ours)		
	MLA	PTA	RTA	MLA	PTA	RTA	MLA	PTA	RTA	MLA	PTA	RTA	MLA	PTA	RTA
Android	73.0	56.6	62.0	76.7	58.4	68.4	63.8	58.9	68.4	97.8	89.4	92.1	**99.6**	**94.6**	**96.4**
BGL	44.4	33.9	30.8	97.0	68.6	78.3	94.0	68.4	77.5	99.4	93.5	95.8	**99.9**	**95.1**	**98.3**
Hadoop	43.9	36.8	34.2	89.5	54.0	58.8	89.0	69.3	85.1	99.4	92.2	97.4	**100.0**	**100.0**	**100.0**
HDFS	95.9	81.3	92.9	90.2	85.7	85.7	**100.0**	**100.0**	**100.0**	99.9	86.7	92.9	**100.0**	**100.0**	**100.0**
Linux	19.4	43.4	42.2	94.9	47.5	49.1	97.3	92.4	93.2	99.7	96.6	96.6	**99.8**	**96.6**	**98.3**
Mac	27.2	21.2	24.9	67.3	43.6	53.4	62.4	48.3	64.5	93.2	74.4	82.1	**96.0**	**77.1**	**85.9**
Thunderbird	19.1	29.9	36.9	92.6	50.6	59.1	88.9	86.8	92.6	98.9	83.3	90.6	**99.9**	**93.3**	**98.7**
Zookeeper	49.8	39.1	36.0	99.0	74.1	86.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
HealthApp	24.1	8.3	34.7	78.9	85.3	85.3	99.9	98.7	98.7	99.9	98.7	98.7	**100.0**	**100.0**	**100.0**
Spark	37.6	50.0	41.7	99.1	60.0	58.3	82.1	48.3	77.8	99.9	97.2	97.2	**100.0**	**100.0**	**100.0**
Windows	69.6	46.3	50.0	98.3	55.4	72.0	97.6	55.9	76.0	99.9	92.3	96.0	**100.0**	**100.0**	**100.0**
OpenSSH	53.4	52.0	50.0	97.6	48.9	84.6	99.9	96.3	96.3	99.9	96.3	96.3	**100.0**	**100.0**	**100.0**
OpenStack	18.0	5.5	39.5	90.7	84.4	88.4	96.9	74.0	88.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Proxifier	52.7	26.9	87.5	**100.0**	**100.0**	**100.0**	96.5	14.3	75.0	99.9	77.8	87.5	**100.0**	**100.0**	**100.0**
HPC	67.2	38.8	41.3	94.7	73.6	84.8	97.5	42.6	87.0	98.6	62.5	97.8	**100.0**	**100.0**	**100.0**
Apache	**100.0**	**100.0**	**100.0**	99.4	83.3	83.3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

Dataset	Generation Time (s)			API Cost (USD)		
	DivLog	AdaICL	LLMLog	DivLog	AdaICL	LLMLog
Android	1.1	1.1	**0.7**	3.5	3.5	**2.1**
BGL	1.4	1.1	**0.8**	3.8	3.7	**2.8**
Hadoop	1.1	1.0	**0.6**	3.7	3.7	**2.0**
HDFS	1.0	1.0	**0.7**	7.6	7.6	**5.1**
Linux	1.2	1.1	**0.7**	3.1	3.1	**2.0**
Mac	1.1	1.1	**0.8**	5.5	5.4	**5.0**
Thunderbird	1.1	1.0	**0.6**	3.3	3.2	**2.7**
Zookeeper	1.1	1.1	**0.3**	3.1	3.1	**2.1**
HealthApp	1.8	1.4	**0.8**	4.4	3.7	**2.3**
Spark	2.1	1.5	**1.4**	6.5	5.7	**3.3**
Windows	2.2	**0.7**	**0.7**	5.3	5.3	**2.6**
OpenSSH	1.1	1.1	**0.7**	7.8	7.8	**2.5**
OpenStack	1.8	1.9	**1.0**	9.8	10.0	**4.2**
Proxifier	0.9	0.9	**0.8**	5.7	5.7	**3.5**
HPC	1.6	0.5	**0.3**	2.6	1.8	**1.3**
Apache	1.8	0.5	**0.4**	2.7	2.4	**1.6**

In-context Learning [VLDBJ'25]: Prompt4Vis

1. Problem

Increasing need of prompting LLMs to transform natural language questions into data visualization queries.

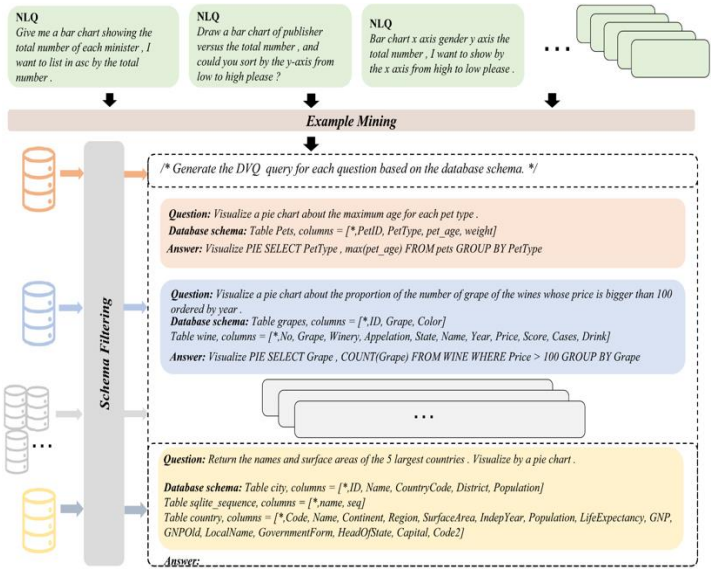


Limitations of Existing Approaches

- Similarity-based example selection provides less demonstrative prompt.
- Expensive costs of providing the whole schema.

2.Key Insight

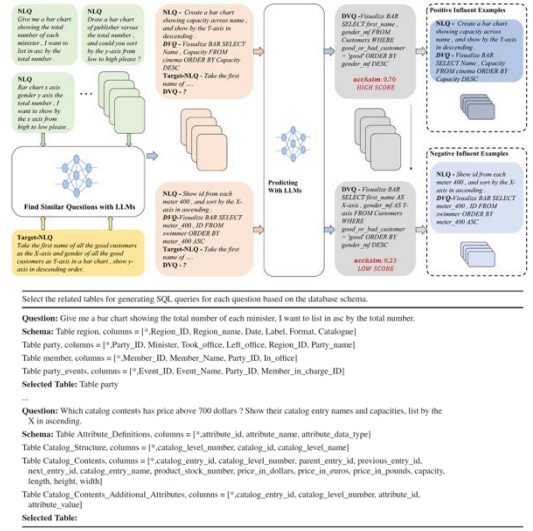
- The **diversity** of examples are necessary to remove redundance.
- The **estimated influence** is crucial for example selection.
- **Filter the irrelevant schemas** for the target query.



3. Method: RAG-Sketch Framework

- **Effective Example Selection Module**: select top-*k* examples that maximize the similarity, diversity and estimated influence (by machine-learning model);
- **Schema Filtering Module**: removes the irrelevant tables by few-shot prompting;

Few-shot examples



PROMPT4VIS: prompting large language models with example mining for tabular data visualization, VLDBJ 2025

Reliable RAG [WWW'26]: GORAG

1. Problem

Many real-world tasks require dynamic text annotation with few annotated data

Round 1 Text: *"All their mods should be stripped of moderator status and be perma-suspended."*

Candidate Label Set: Round 1
- Threatening Language
- Anti-human
- Neutral

Round 2 Text: *"Old people are lazy, and their children have to clean up their messes."*

Candidate Label Set: Round 2
- Identity Hatred
- Identity Insults
- Identity Counter Speech

Label sets change overtime. Previous decisions may become misaligned

2. Limitations

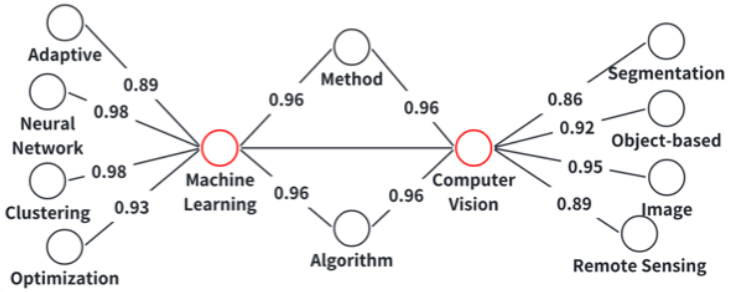
Directly applying LLMs suffer from **high cost** and context limitation

SFT-based methods suffer from **high human labeling cost**

Previous classifications may **misalign with current label set**

3. Method: Keyword-Label Semantic Graph

Key insight: memorize semantic correlations between text keywords and labels, not classification results.

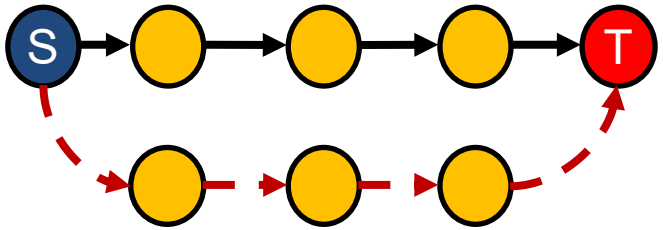


A. Extract text keywords as graph nodes | **B**. Link keywords to labels as edges | **C**. Assign edge weights by **semantic relevance** | **D**. Update graph after each classification batch

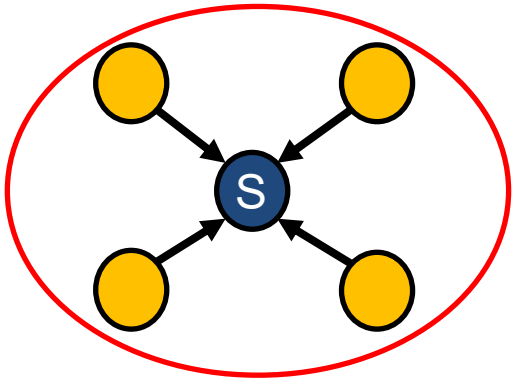
Reliable RAG [ICDE'26]: AGRAG

1. Limitations

Shortest path retrieval **misses useful nodes**.



Neighborhood Retrieval produces **incoherent trajectory**



2.Key Insight

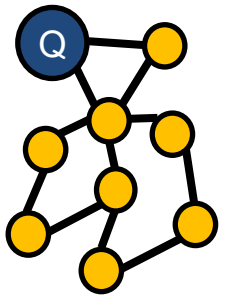
Use more inclusive graph structures for **multi-source, multi-hop** reasoning.

Hybrid Retrieval
for coarse-grained knowledge.

Min-Cost Max-Influence Retrieval
for fine-grained knowledge.

3. Method

Query & KG

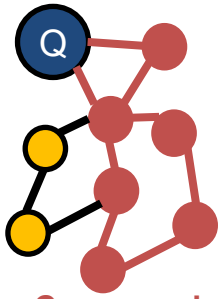


Node Influence: Higher=more query related

Hybrid Retrieval

MCMI Retrieval

Retrieved Subgraph



Converged Subgraph

Edge Cost: Lower=less noise

Retrieve the subgraph once the process converges.

Wang, Yubo, et al. “AGRAG: Advanced Graph-based Retrieval-Augmented Generation for LLMs.” ICDE 2026

Reliable RAG [ICLR'26]: VoG

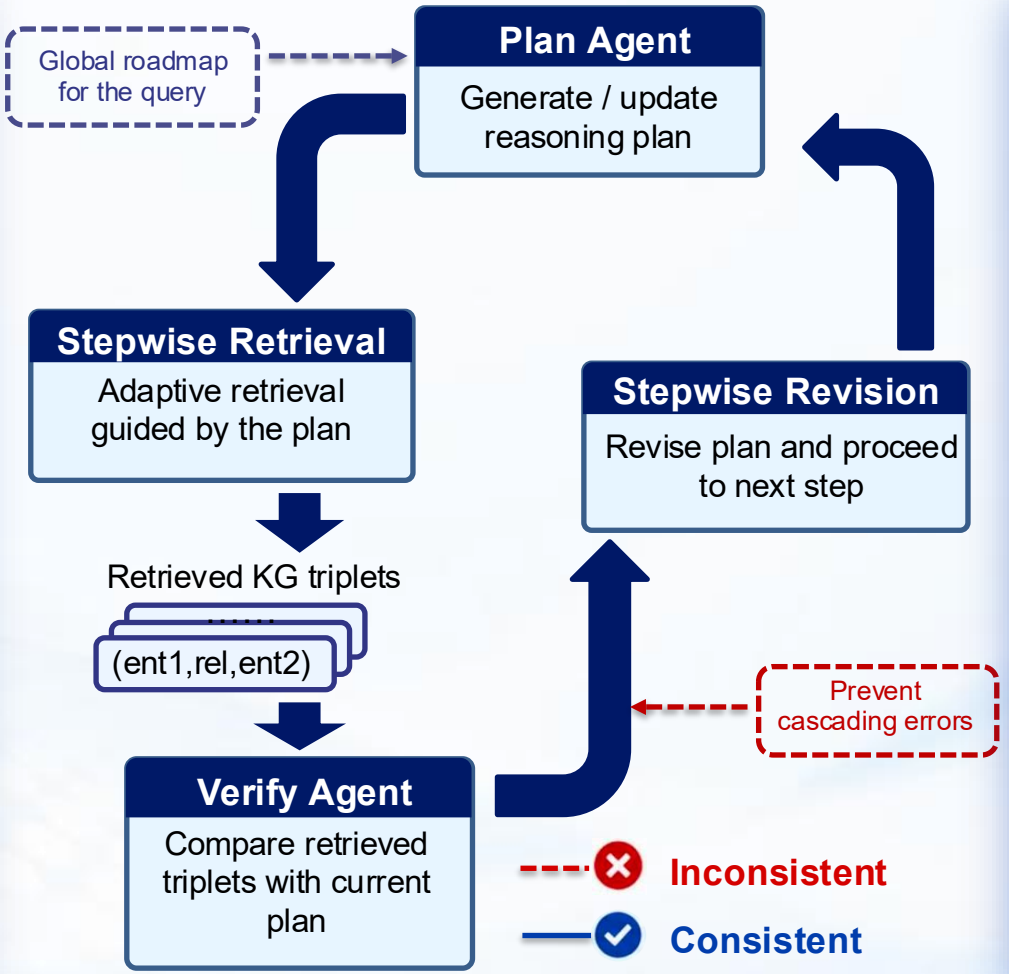
Problem / Insights

- **Error accumulation**
- **Limited scope of reasoning**

Flexible
Plan agent dynamically guide retrieval and adjust reasoning depth

Grounded
Stepwise verification continuously correct errors

Contextual-aware
KG evidence and contextual information revise the reasoning plan



Context Selector

Local
Current-step triplets

Lookahead
Current-step triplets + next-step relations

Global
All triplets + full reasoning plan

UCB- based context selection

Reward
- Local coherence
- answer stability

Revise plan with selected context, then execute from t+1 onward

VoG: Enhancing LLM Reasoning through Stepwise Verification on Knowledge Graphs, ICLR 2026

Summary

Efficient and Reliable Data-Centric Machine Learning Lifecycle

Data Preparation

Data Augmentation

SSIN	SIGMOD '23

Data Cleaning

KBPearl	VLDB '20
Triple-d	ICDE '24
OpenMEL	VLDB '25

Log Identification

MaidLog	ICDE '26

Model Training

Data Replay

ESDR	ICDE '24

Coreset Selection

Camel	SIGMOD '22
ERCA	Under Review

Cache and Batch Technique

KV Cache

APT-Serve	SIGMOD '25
Probe & Fetch	WWW '26

Adaptive Batch

AGRAG	ICDE '26
RAG-Sketch	SIGMOD '26
MGRAG	VLDB '26 R

Reliable Generation

Context Learning

Prompt4Vis	VLDBJ '25
LLMLog	VLDB '25

Reliable RAG

GORAG	WWW '26
VoG	ICLR '26
Probe & Fetch	WWW '26

Summary

High-Quality Data + Coreset Training + Efficient Caching + Reliable RAG

Fundamental Techniques

Core innovations in research

- **Hybrid Caching**
  - KV Cache & Hidden Cache
- **Sparsification**
  - Online Sparse Prefilling
  - Dynamic KV Pruning
- **Weighted Coresets**
  - One-pass update selection
- **Graph-based Retrieval**
  - MCMI Subgraph Matching
  - Maximum Spanning Tree

Design Choices

Workload & Reliability trade-offs

- **Request Scheduling**
  - Adaptive Runtime Schedule
  - FCFS vs. Value-based
- **Buffer Management**
  - Merge-and-Reduce (MR)
  - Entropy-based selection
- **Reasoning Depth**
  - Stepwise Verification (VoG)
  - Multi-round Annotation
- **Data Format**
  - Structured vs. non-structured

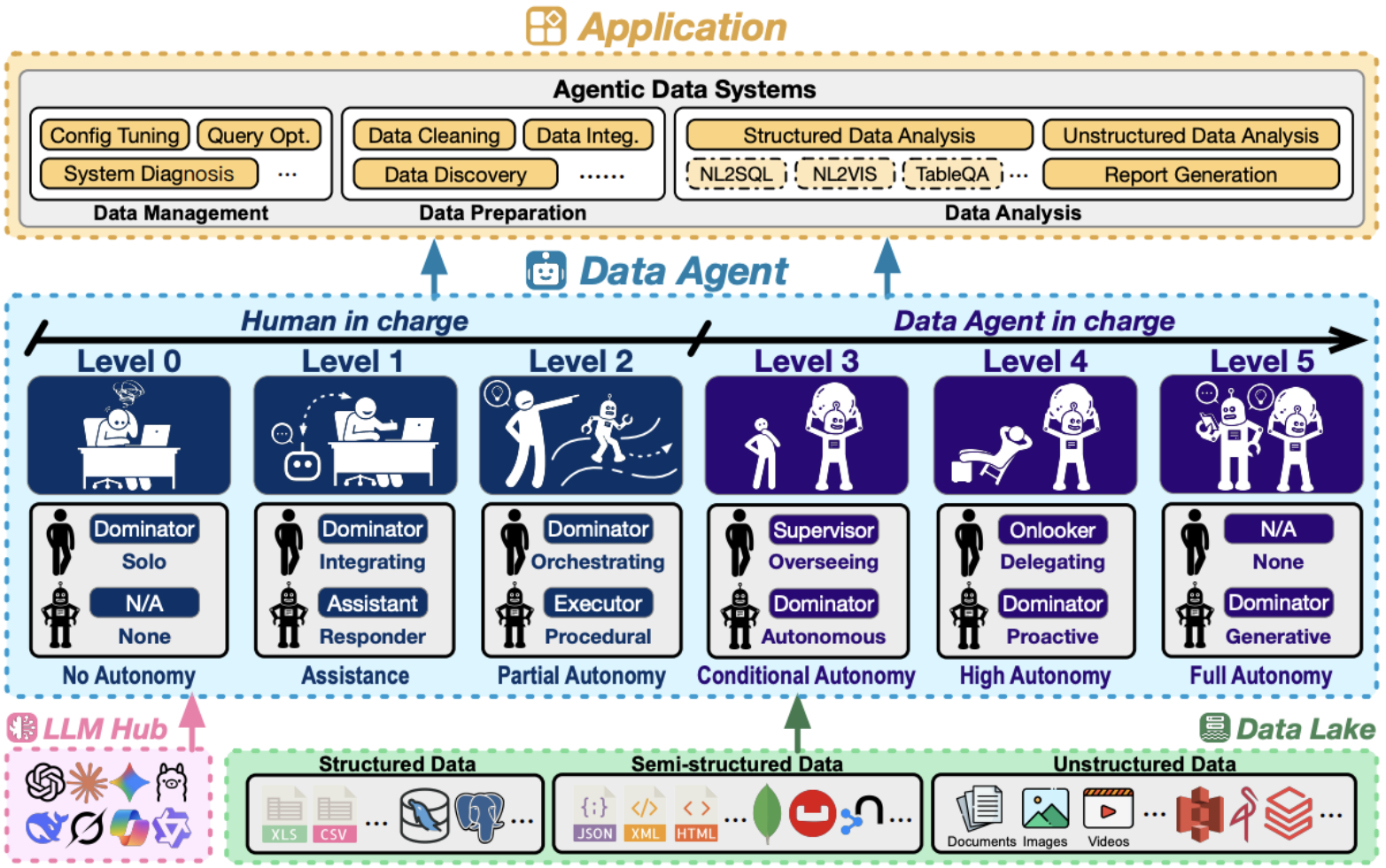
Optimization

Performance & Cost objectives

- **Throughput Target**
  - Effective throughput 8.8x
- **Latency Constraints**
  - TTFT (First Token)
  - P99 TBT (Subsequent)
- **Resource Budget**
  - Limited GPU Memory usage
  - Human Annotation budget
- **Efficiency Metrics**
  - Sample Ratio (30%-100%)
  - Token cost minimization

Autonomous Agentic Data Workflows

Evolving from static, manual data preparation to dynamic, agent-driven data curation pipelines.



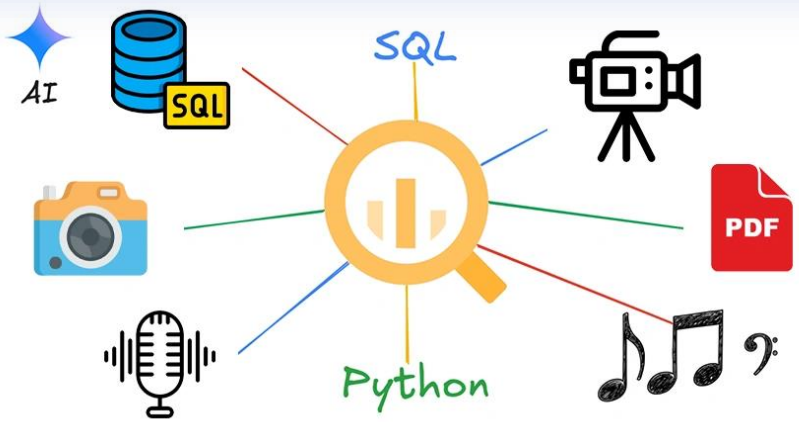
Multimodal and Spatial-Temporal Frontier

Expanding AI capabilities to interact with complex physical environments and the digital Metaverse.

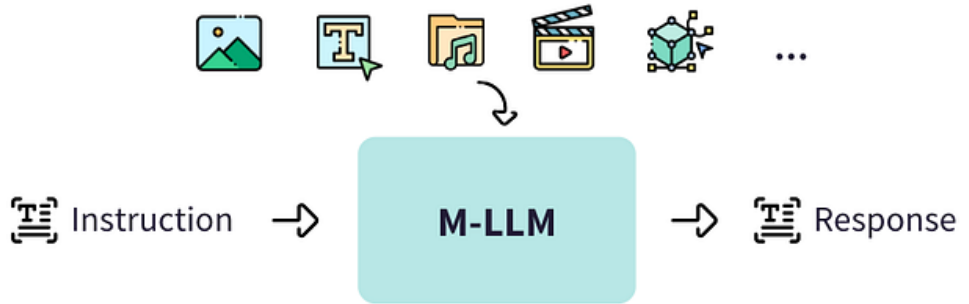
Data-Centric Multimodal Spatial-Temporal Data Curation

Data-Driven MLLM Alignment for Reliable Multimodal Reasoning

Multi-Modal Data Curation



Multi-Modal Alignment



Papers

SIGMOD:

1.Camel: Managing Data for Efficient Stream Learning. [SIGMOD'22]
2.SSIN: Self-Supervised Learning for Rainfall Spatial Interpolation. [SIGMOD'23]
3.Apt-Serve: Adaptive Request Scheduling on Hybrid Cache for Scalable LLM Inference Serving. [SIGMOD'25]
4.Skyline Retrieval meets Set-Cover Chunk Merging: A Cost-Effective RAG-Sketch for Long-Context LLM QA. [SIGMOD'26]

VLDB & VLDBJ:

1.KBPearl: a knowledge base population system supported by joint entity and relation linking. [VLDB'20]
2.OpenMEL: Unsupervised Multimodal Entity Linking Using Noise-Free Expanded Queries and Global Coherence. [VLDB'25]
3.LLMLog: Advanced Log Template Generation via LLM-driven Multi-Round Annotation. [VLDB'25]
4.Prompt4vis: prompting large language models with example mining for tabular data visualization. [VLDBJ'25]
5.Semantic Subgraph Matching and Graph-Aware Caching for Multimodal Retrieval-Augmented Generation [VLDB'26 Revision]

ICDE:

1.Effective Data Selection and Replay for Unsupervised Continual Learning. [ICDE'24]
2.Triple-d: Denoising distant supervision for high-quality data creation [ICDE '24]
3.AGRAG: Advanced Graph-based Retrieval-Augmented Generation for LLMs. [ICDE'26]
4.Efficient Zero-shot and Label-free Log Anomaly Detection for Resource-constrained Systems [ICDE'26]

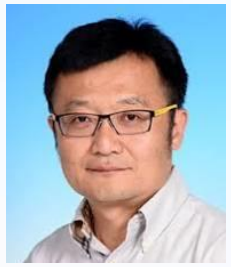
ICLR:

1.VoG: Enhancing LLM Reasoning through Stepwise Verification on Knowledge Graphs. [ICLR'26]

WWW:

1.Graph-based Online Retrieval Augmented Generation for Dynamic Few-shot Social Media Text Classification. [WWW'26]
2.Probe-and-Fetch: Dynamic KV Cache Pruning for Accelerated Long-Context Inference in Web-Scale AI Search." [WWW'26]

Team Members



Lei Chen

Yanyan Shen

Jason Zhang

Xueling Lin

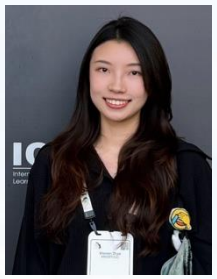
Yongqi Zhang

Haoyang Li

Yiming Li

Yubo Wang

Xinyi Zhu



Wenxin Zhao

Zuohan Wu

Tianhao Tang

Fei Teng

Shihong Gao


Thanks
for your listening!
We are recruiting ...
Be Part of Information Hub, HKUST(GZ)
Apply Now