

KubeEdge云边协同实践：大语言模型云边协同推理benchmark套件

胡时京 复旦大学 KubeEdge

范彧 北京航空航天大学 KubeEdge

Content 目录

- 01** 大模型云边协同背景
- 02** KubeEdge大模型云边协同推理新范式
- 03** 基于KubeEdge-lanvs的大模云边协同实践
- 04** 开源成果分享

Part 01

大模型云边协同背景



为什么我们需要大模型云边协同？

- 每年LLM API开销超过100亿人民币
- 云端LLM API开销仍然较为昂贵

Latest models

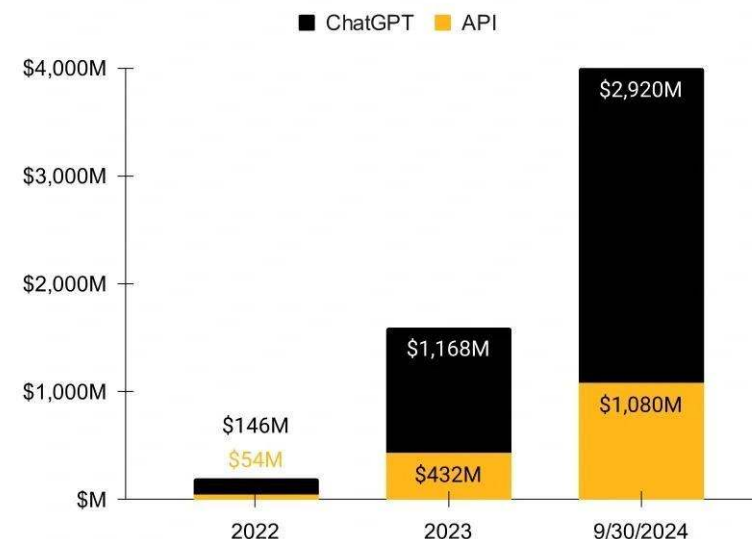
Text tokens

Price per 1M tokens: Batch API price: ☐

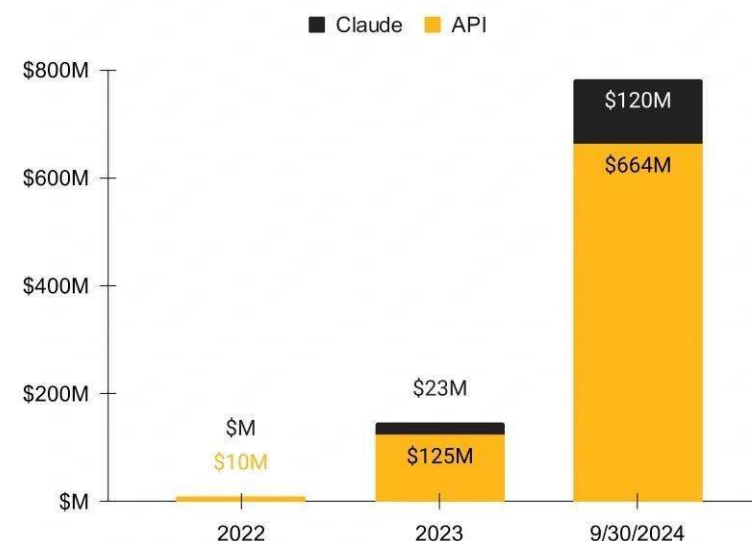
Model	Input	Cached input	Output
gpt-4.5-preview ↳ gpt-4.5-preview-2025-02-27	\$75.00	\$37.50	\$150.00
gpt-4o ↳ gpt-4o-2024-08-06	\$2.50	\$1.25	\$10.00
gpt-4o-audio-preview ↳ gpt-4o-audio-preview-2024-12-17	\$2.50	-	\$10.00
gpt-4o-realtime-preview ↳ gpt-4o-realtime-preview-2024-12-17	\$5.00	\$2.50	\$20.00
gpt-4o-mini ↳ gpt-4o-mini-2024-07-18	\$0.15	\$0.075	\$0.60
gpt-4o-mini-audio-preview ↳ gpt-4o-mini-audio-preview-2024-12-17	\$0.15	-	\$0.60
gpt-4o-mini-realtime-preview ↳ gpt-4o-mini-realtime-preview-2024-12-17	\$0.60	\$0.30	\$2.40
o1 ↳ o1-2024-12-17	\$15.00	\$7.50	\$60.00
o3-mini ↳ o3-mini-2025-01-31	\$1.10	\$0.55	\$4.40



OpenAI
Revenue mix



ANTHROPIC
Revenue mix



为什么我们需要大模型云边协同？

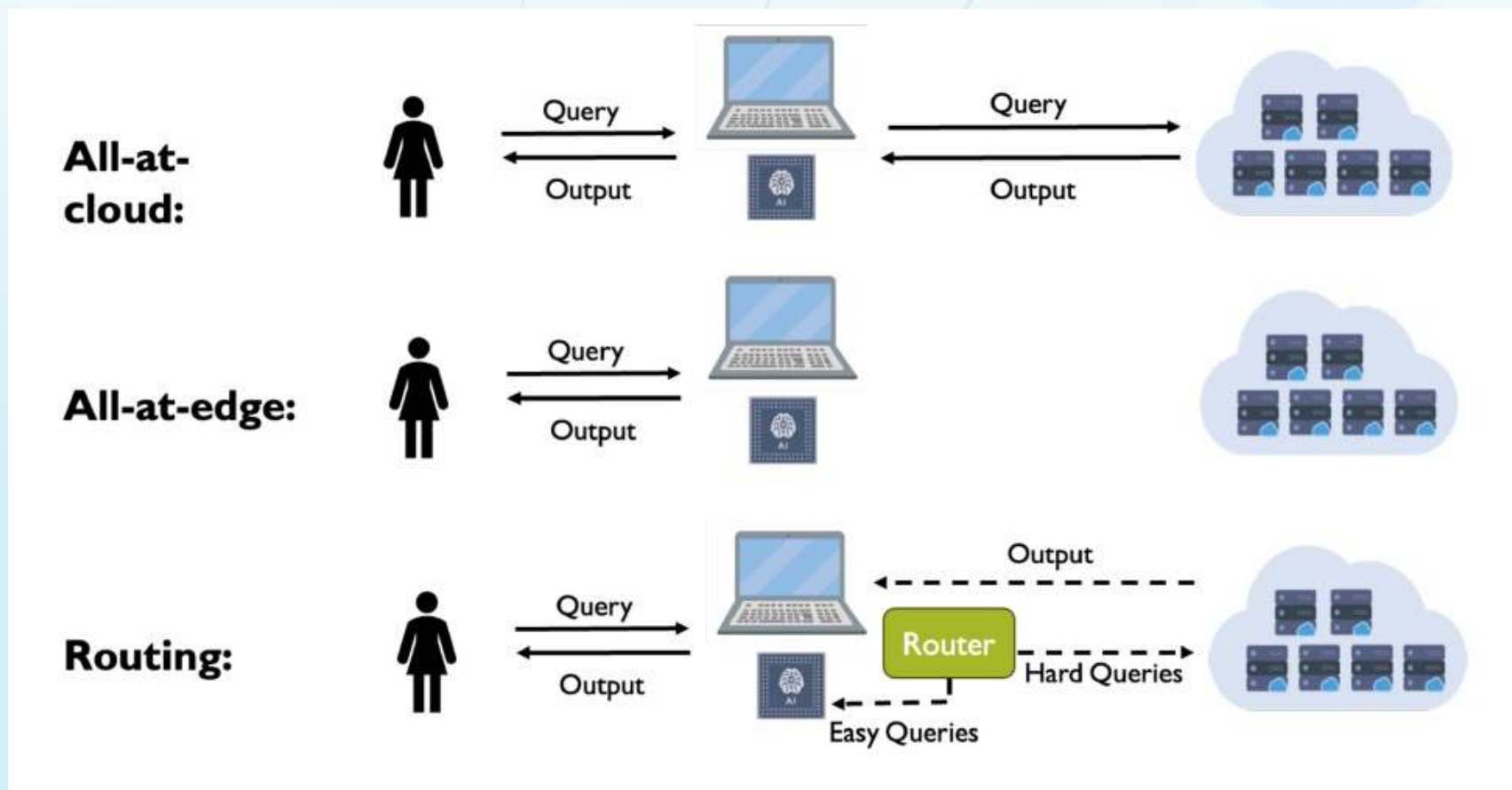
- 边缘端可以部署的LLM模型能力越来越强
- 在较难任务上边缘端LLM模型与云端LLM模型还有较大差距
- 满血Deepseek-r1 671b在边缘部署资源开销较大

模型大小	参数量	显存需求 (GPU)	CPU 和内存需求	适用场景
1.5B	15亿	2-4 GB	8 GB 内存	低端设备，轻量推理
7B	70亿	8-12 GB	16 GB 内存	中端设备，通用推理
8B	80亿	10-16 GB	16-32 GB 内存	中高端设备，高性能推理
14B	140亿	16-24 GB	32 GB 内存	高端设备，高性能推理
32B	320亿	32-48 GB	64 GB 内存	高端设备，专业推理
70B	700亿	64 GB+	128 GB 内存	顶级设备，大规模推理
671B	6710亿	多 GPU (80 GB+)	256 GB+ 内存	超大规模推理，分布式计算

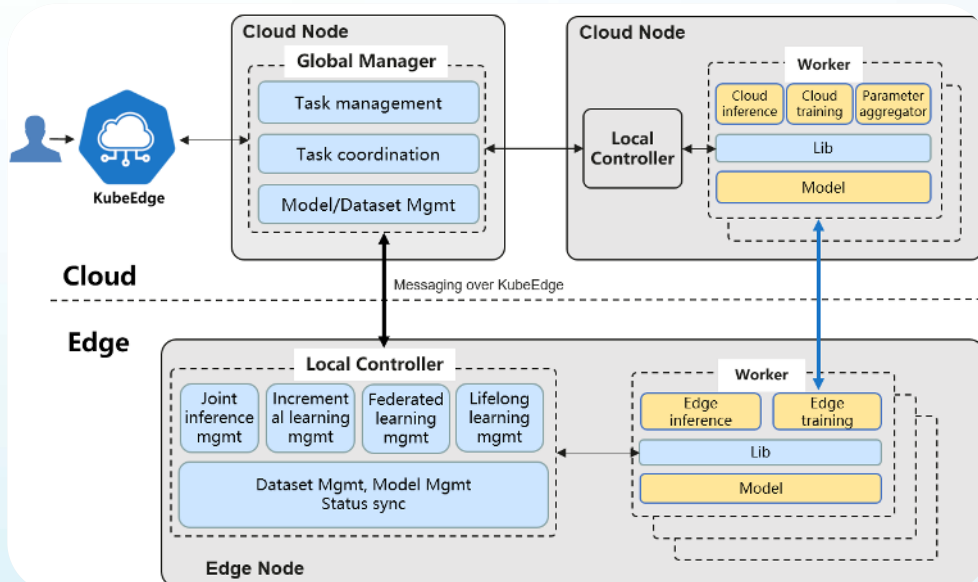
Model	AIME 2024 pass@1	AIME 2024 cons@64	MATH-500 pass@1	GPQA Diamond pass@1	LiveCodeBench pass@1	CodeForces rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
o1-mini	63.6	80.0	90.0	60.0	53.8	1820
QwQ-32B-Preview	44.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2	1691

为什么我们需要大模型云边协同？

- 节省云端LLM API调用成本（**每年超过100亿人民币的市场**）
- 提高边侧LLM回答准确率（利用云侧LLM更强的能力解决更难的问题）



为什么选择KubeEdge作为大模型云边协同基础设施



<https://github.com/kubeedge/sedna>

首个分布式协同AI开源项目Sedna

基于KubeEdge提供的边云协同能力，支持现有AI类应用无缝下沉到边缘

为分布式协同机器学习服务

- ✓ 降低构建与部署成本
- ✓ 提升模型性能
- ✓ 保护数据隐私

基础框架

- ✓ 数据集管理
- ✓ 模型管理
- ✓

训练推理框架

- ✓ 协同推理
- ✓ 增量学习
- ✓ 联邦学习
- ✓ 终身学习

兼容性

- ✓ 主流AI框架
- ✓ 模块算法
- ✓ 可扩展算法接口
- ✓

为什么选择KubeEdge作为大模型云边协同基础设施



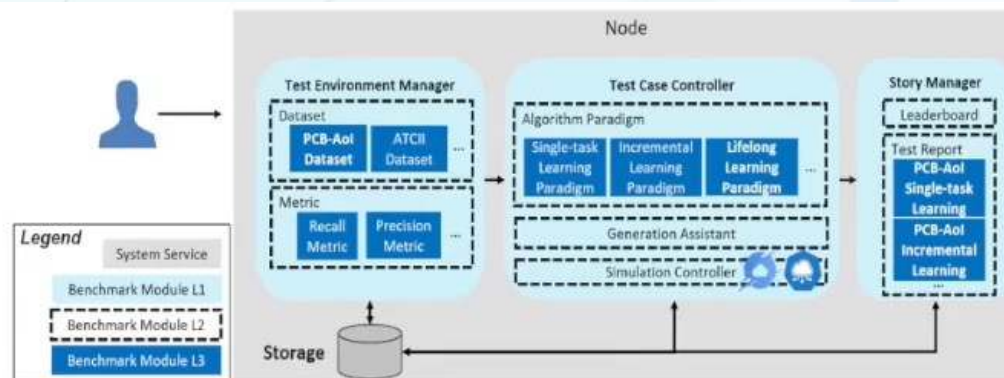
核心痛点

业务数据集及其配套算法难以获取

全场景多范式测试成本高

封闭测试环境难以跟上各类新业务孵化

个性化场景的测试用例准备繁琐



丰富AI生态，开箱即用

数据集与配套算法，覆盖开发5+流程，
零改造开箱即用

全场景灵活切换

用例管理统一不同架构与接口，
同一套工具兼容5+场景范式

可扩展开放工具链

环境管理自定义数据集与指标
告别封闭守旧的测试环境

低代码生成测试用例

用例管理辅助生成测试用例，
简单配置即可降低繁琐重复编程

项目地址：<https://github.com/kubeedge/ianvs>

欢迎关注本项目，持续获得第一手独家公开数据集与完善基准测试配套

技术验证时间 半年 ➡ 一个月，5倍研发效率提升

Part 02

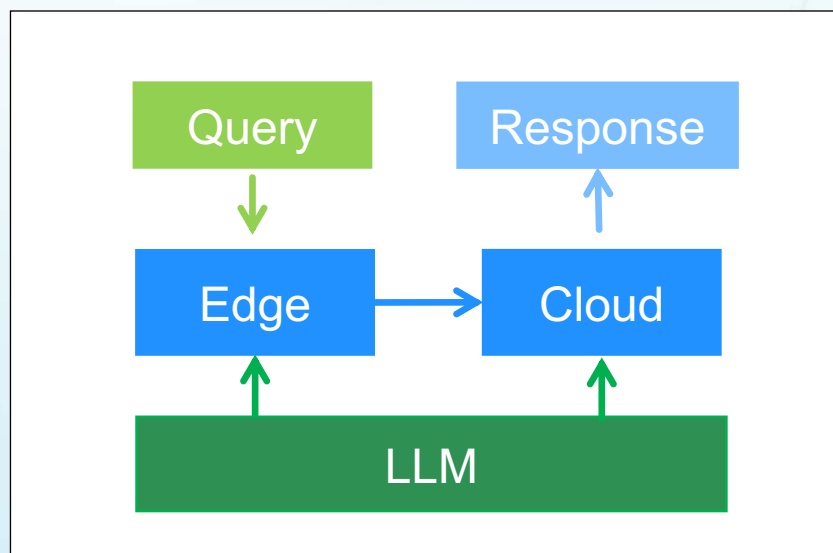
KubeEdge大模型云边协同推理新范式

AI



2.1 候选的云边协同推理方式

范式1: 模型切片

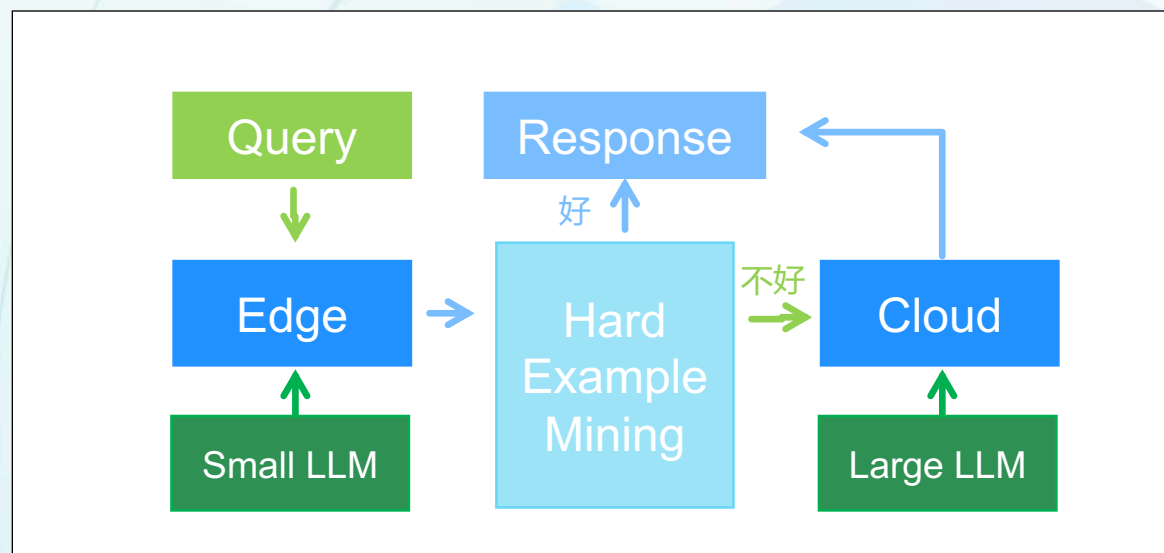


◆ 异构网络实现模型部署

解决隐私问题；带宽需求高；首字时延高

相关工作：EdgeShard¹；PerLLM²

范式2: 先推理后挖掘难例



◆ CV 场景下常见的协同策略

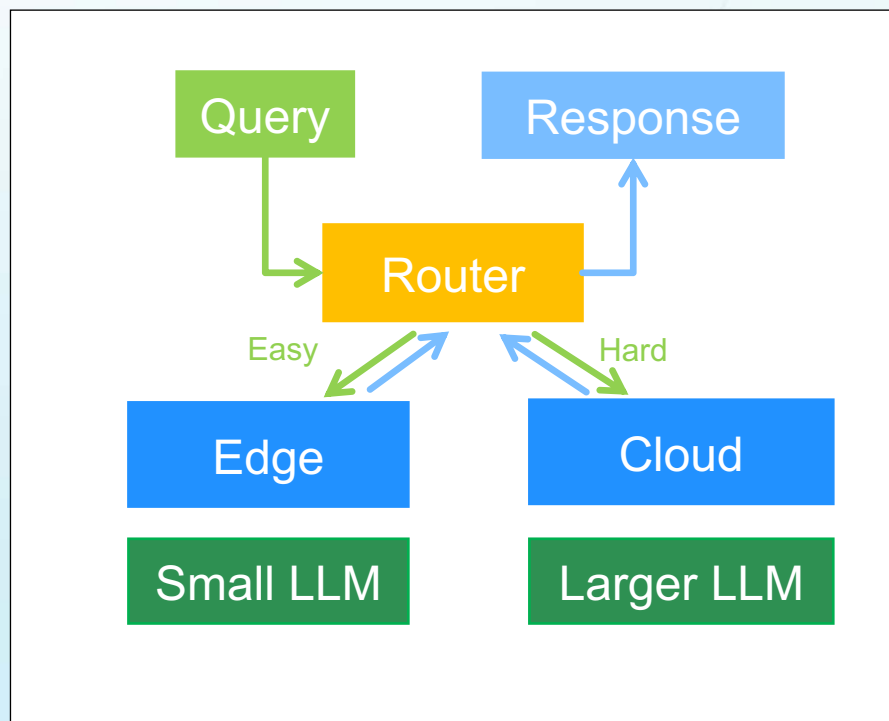
缓解隐私问题；带宽需求小；首字时延非常高

1. Zhang, Mingjin, et al. *EdgeShard: Efficient LLM Inference via Collaborative Edge Computing*. arXiv:2405.14371

2. Yang, Zheming, et al. *PerLLM: Personalized Inference Scheduling with Edge-Cloud Collaboration for Diverse LLM Services*. arXiv:2405.14636

2.2 较优的云边协同推理方式

范式3: 查询路由 (Query-Routing)



核心思想

识别简单的请求并将其路由到边缘模型

示例

简单请求：求 $5 \sin\left(\frac{\pi}{2}\right)$

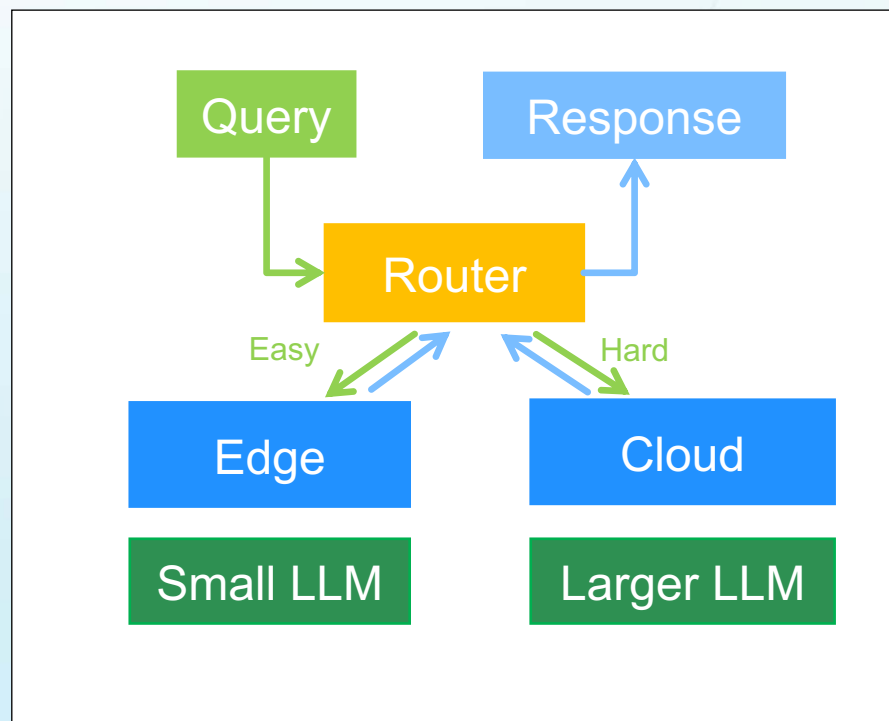
复杂请求：给定区域 D ，求 $\iint_D \sqrt{1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} dx dy$

简单请求：将下面这段翻译为英文：欢迎参加 KCD！

复杂请求：按照正式会议的翻译习惯，将 KCD 的会议记录信达雅地翻译为英文。

2.2 较优的云边协同推理方式

范式3: 查询路由 (Query-Routing)



优势

在不降低回复质量的前提下，查询路由机制可以：

- **减少使用成本**：
对于模型用户，减少顶级 API 使用开销；
对于模型厂商，合理调配模型降低推理成本
- **降低首字时延**：边端模型几乎无传播时延
- **缓解隐私问题**：仅有部分请求需要上云

相关工作：

Hybrid LLM³、RouteLLM⁴、Prompt2Leaderboard⁵


3. Ding, Dujian, et al. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. ICLR 2024.

4. Ong, Isaac, et al. RouteLLM: Learning to Route LLMs with Preference Data. ICLR 2025.

5. Frick, Evan, et al. Prompt-to-Leaderboard. arXiv:2502.14855

2.3 Query Routing 的效果



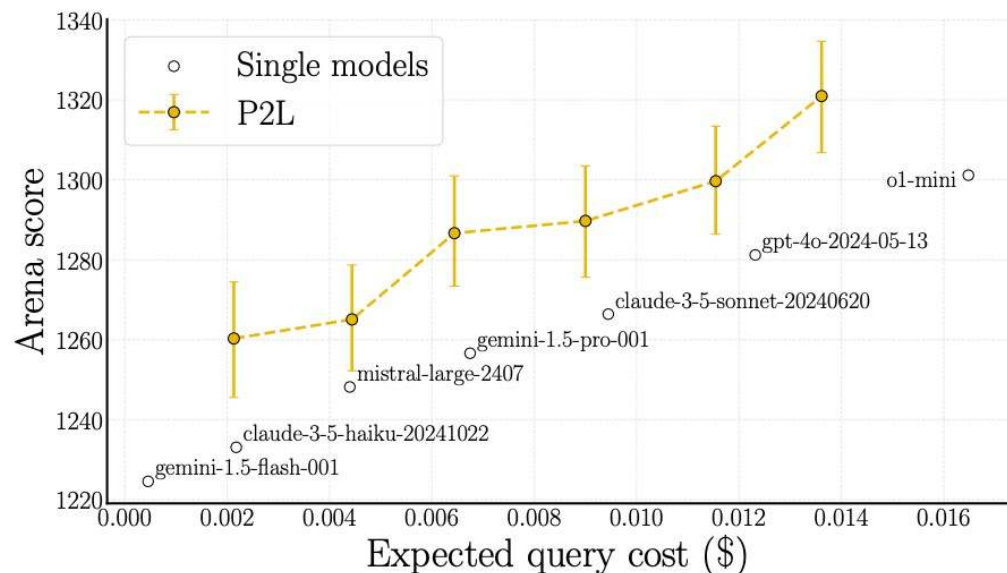
lmarena.ai (formerly lmsys.org) 

@lmarena_ai

用例 1: 最佳路由

如果我们知道每个提示中哪些模型是最好的, 那么最佳路由就变得简单了!

- 性能: P2L-router (experimental-router-0112) 在 2025 年 1 月的 Chatbot Arena 中排名第一, 得分为 1395。(比最佳模型候选高出 20 分)
- P2L 还支持在给定成本约束的条件下进行路由, 以实现帕累托前沿



Rank* (UB)	Rank (StyleCtrl)	Model	Arena Score	95% CI	Votes	Organization	License
1	2	Grok-3-Preview-02-24	1407	+7/-7	7580	xAI	Proprietary
1	1	GPT-4.5-Preview	1404	+7/-9	6024	OpenAI	Proprietary
3	6	Gemini-2.0-Flash-Thinking-Exp-01-21	1384	+5/-5	19837	Google	Proprietary
3	3	Gemini-2.0-Pro-Exp-02-05	1380	+4/-4	17695	Google	Proprietary
3	2	ChatGPT-4o-latest (2025-01-29)	1375	+4/-5	19587	OpenAI	Proprietary
6	4	DeepSeek-R1	1361	+5/-6	10474	DeepSeek	MIT
6	10	Gemini-2.0-Flash-001	1355	+4/-5	15416	Google	Proprietary
6	3	o1-2024-12-17	1353	+4/-4	22010	OpenAI	Proprietary
9	10	Gemma-3-27B-it	1339	+9/-11	3870	Google	Gemma
9	10	Qwen2.5-Max	1338	+5/-5	14258	Alibaba	Proprietary
9	7	o1-preview	1335	+4/-4	33195	OpenAI	Proprietary
9	10	o3-mini-high	1328	+6/-5	11409	OpenAI	Proprietary

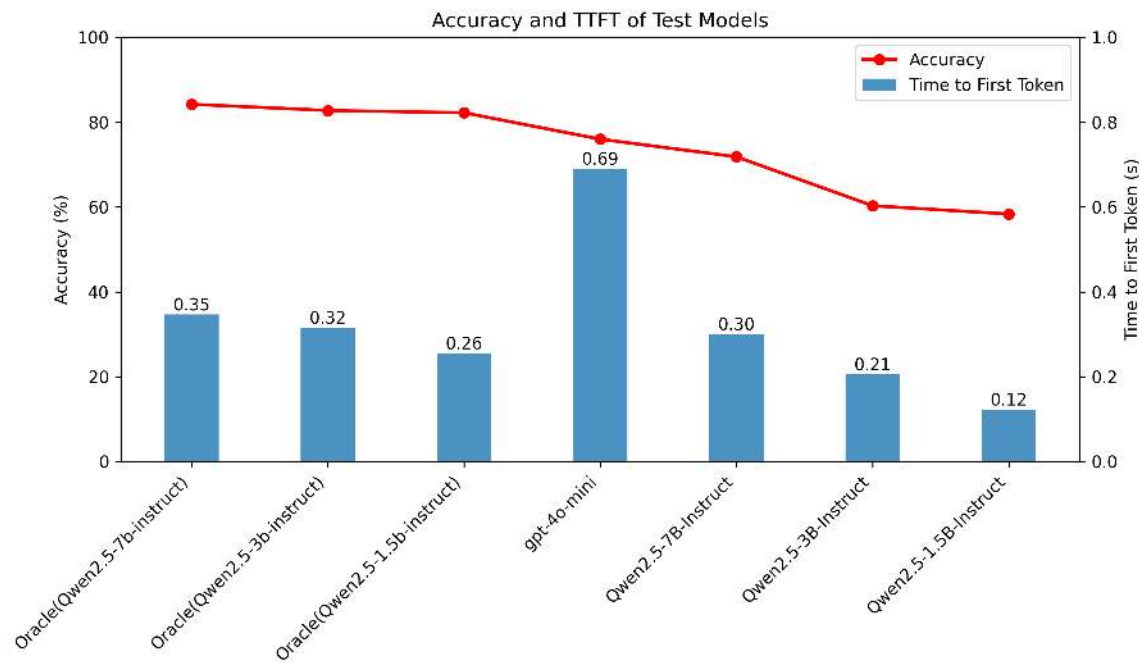
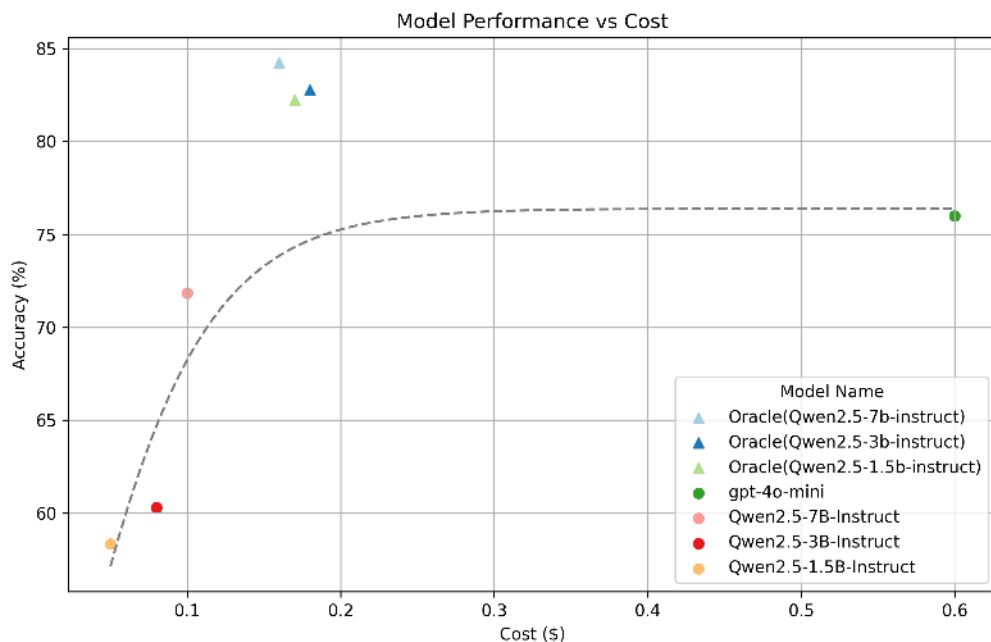
引入查询路由后构成的新系统：1. 给定预算情况下，能够获得更高质量的回复

2. 在 ChatBot Arena 中获得了 1395 分，超越榜单上原有的所有模型

2.3 Query Routing 的效果

使用理想分类器在 MMLU (5-shot) 测试集上进行测试

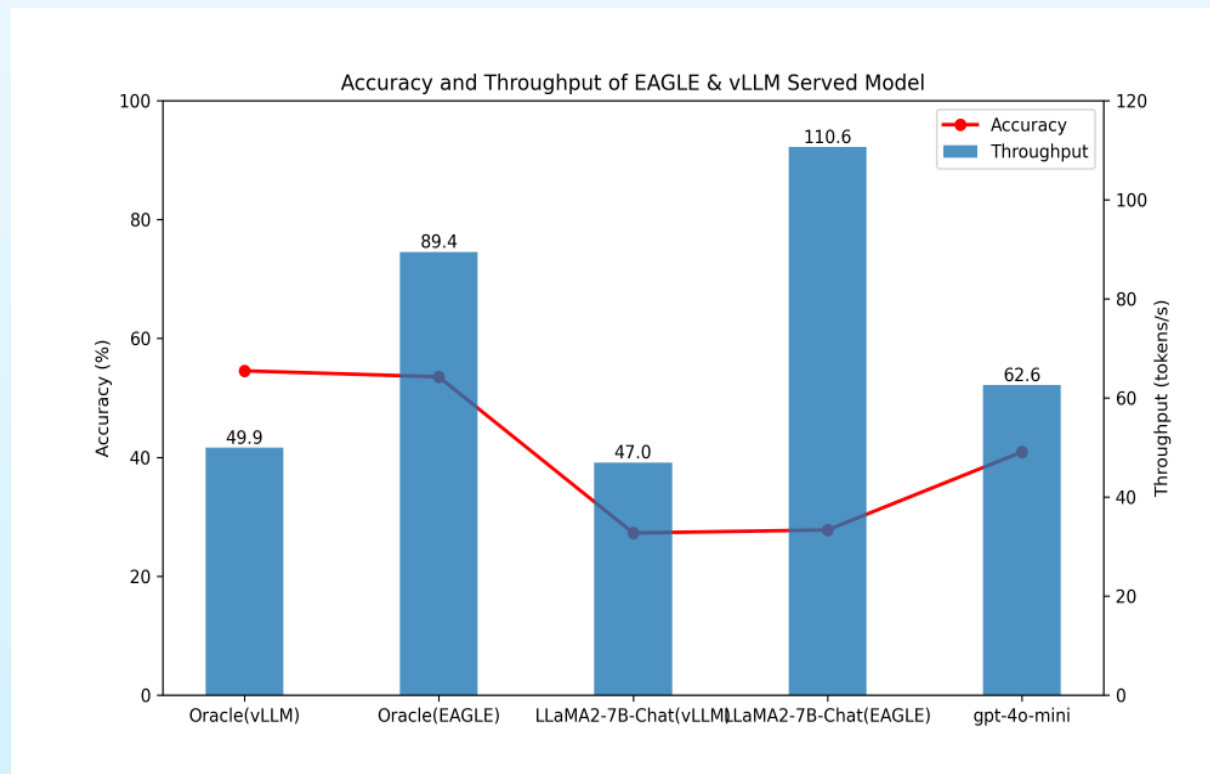
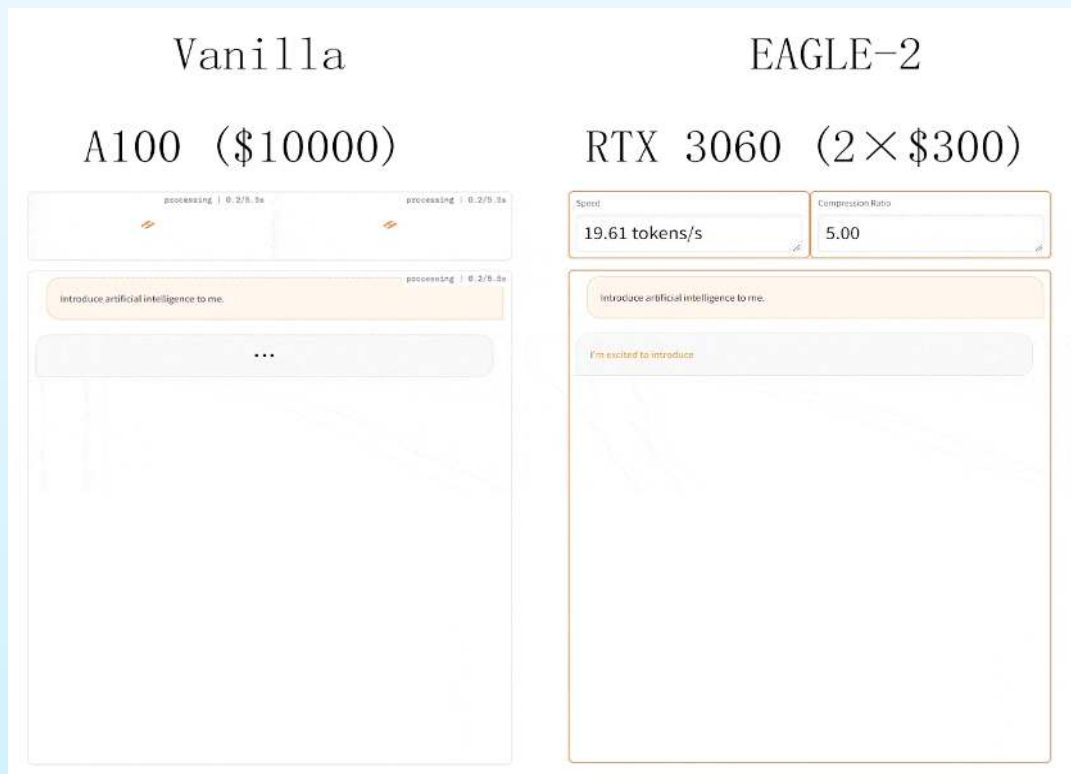
- 当 Qwen2.5 系列模型与 gpt-4o-mini 进行理想协作时，87% 的请求可交给边缘侧的 Qwen 完成
- 实现 12.38% (相比 Qwen 自身) 和 8.23% (相比 gpt-4o-mini) 的精度提升；并降低至少 50% 的首字时延



2.4 进一步加速 - 投机解码 (Speculative Decoding)

核心思想：使用一个 Draft Model 快速预测多个 Token，随后使用 Large Model 进行并行验证

- 在 Query-Routing 后，**进一步提升边缘侧/云侧的推理速度**，提升系统的吞吐量，**为用户提供更好的体验**；
- LLaMA2-7B-Chat 实验：EAGLE 可以显著加速 LLM 的生成 (**2.35x**)；进行理想协作时，也能为系统带来 **1.79x** 的提升

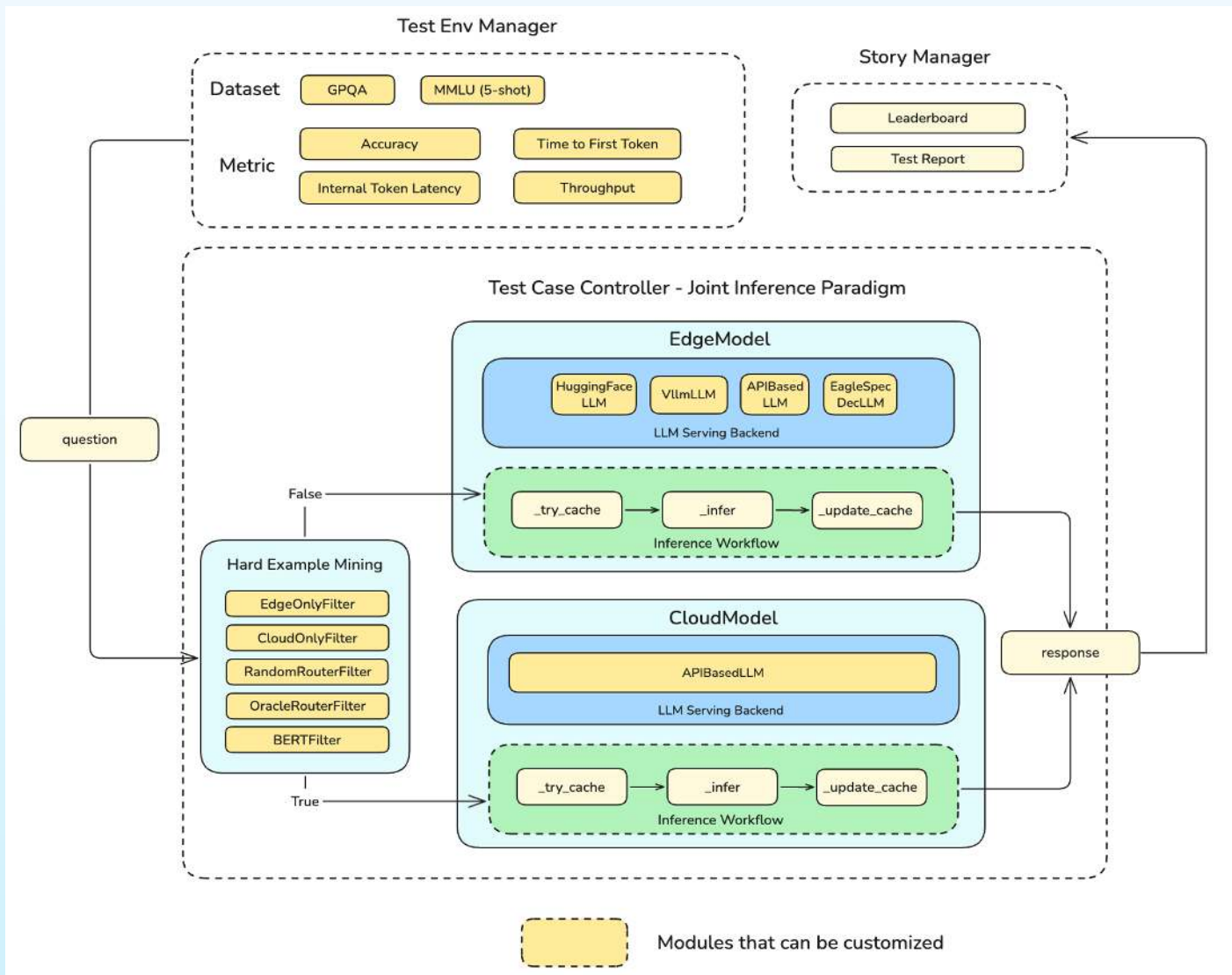


Part 03

基于 KubeEdge-lanvs 的大模型云边协同实践



3.1 基于查询路由的 LLM 云边协同推理 Benchmark 架构



架构特色

- **丰富的指标统计**
 - 正确率 Accuracy (%)
 - 首字时延 TTFT (s)
 - 吞吐量 Throughput (Token/s)
- **多样的 Inference 支持**
 - Offline: transformers, vLLM, EAGLE
 - Online: OpenAI API Client
 - 具有缓存机制, 支持断点续测
- **多样的 Router 示例**
 - EdgeOnly, CloudOnly, Random
 - BERT-based Router
 - 最优路由 OracleRouter



3.2 环境准备

```
# Clone Ianvs Repo
git clone https://github.com/kubeedge/ianvs.git
cd ianvs

# 安装 KubeEdge Sedna
pip install examples/resources/third_party/sedna-0.6.0.1-py3-
none-any.whl

# 安装 KubeEdge Ianvs
pip install -r requirements.txt
python setup.py install

# 安装 Query-Routing 的依赖
pip install -r examples/cloud-edge-collaborative-inference-
for-llm/requirements.txt
```



3.3 测试数据集准备

KubeEdge Ianvs 对数据集格式进行了约定，至少需要准备以下两个文件：

1. *metadata.json*

```
{  
  "dataset": "MMLU",  
  "description": "Measuring Massive Multitask Language Understanding by Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt (ICLR 2021).",  
  "level_1_dim": "single-modal",  
  "level_2_dim": "text",  
  "level_3_dim": "Q&A",  
  "level_4_dim": "general"  
}
```

2. *test.jsonl*

```
{"query": "Question: Find the degree for the given field extension  $Q(\sqrt{2}, \sqrt{3}, \sqrt{18})$  over  $Q$ .  
A. 0  
B. 4  
C. 2  
D. 6", "response": "B",  
"explanation": "", "level_1_dim": "single-modal", "level_2_dim": "text",  
"level_3_dim": "knowledge Q&A", "level_4_dim": "abstract_algebra"}
```



3.4 模型参数配置

1. EdgeModel 及其开放参数

参数名	类型	描述	默认值
model	str	模型名称	Qwen/Qwen2-1.5B-Instruct
backend	str	推理框架	huggingface
temperature	float	温度, 0~2	0.8
top_p	float	核采样参数	0.8
max_tokens	int	最大补全 Token 数	512
repetition_penalty	float	重复惩罚因子	1.05
tensor_parallel_size	int	Tensor Parallel 数量 (仅适用于vLLM)	1
gpu_memory_utilization	float	GPU 显存利用率 (仅适用于vLLM)	0.9
draft_model	str	投机解码草稿模型 (仅适用于EAGLE)	-

2. CloudModel 及其开放参数

参数名	类型	描述	默认值
model	str	模型名称	gpt-4o-mini
temperature	float	采样温度	0.8
top_p	float	核采样参数	0.8
max_tokens	int	最大补全 Token 数	512
repetition_penalty	float	重复惩罚因子	1.05

3. Router 及其开放参数

Router 类型	描述	开放参数
EdgeOnly	将所有请求都路由到EdgeModel	-
CloudOnly	将所有请求都路由到CloudModel	-
OracleRouter	理想的最优路由	-
BERTRouter	使用 BERT 对请求进行分类路由	model, threshold
RandomRouter	随机地将请求路由到Edge/Cloud	threshold

```
export OPENAI_BASE_URL="https://api.openai.com/v1"
export OPENAI_API_KEY=sk_XXXXXXXXXX
```

对于 BERTRouter , 可以下载 RouteLLM 提供的
[😊 routellm/bert_gpt4_augmented](https://github.com/RouteLLM/routellm/bert_gpt4_augmented) 模型进行体验



3.5 运行评测

```
ianvs -f examples/cloud-edge-collaborative-inference-for-llm/benchmarkingjob.yaml
```

控制台将实时打印评测状态：

```
(ianvs) (base) root@autodl-container-749c4db1bf-f7c84a1i:~/autodl-imp/ianvs# python benchmarking.py -f examples/cloud-edge-collaborative-inference-for-llm/benchmarkingjob.yaml 2>&1 | tee d
emo.log
[2025-03-12 06:36:36,985] edge_model.py(43) [INFO] - {'model': 'Qwen/Qwen2.5-1.5B-Instruct', 'backend': 'vllm', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.0
5, 'tensor_parallel_size': 4, 'gpu_memory_utilization': 0.9, 'use_cache': True}
[2025-03-12 06:36:36,985] cloud_model.py(34) [INFO] - {'model': 'gpt-4o-mini', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.05, 'use_cache': True}
[2025-03-12 06:36:37,480] joint_inference.py(73) [INFO] - Loading dataset
[2025-03-12 06:36:38,451] hard_sample_mining.py(30) [INFO] - USING OracleRouterFilter
[2025-03-12 06:36:38,452] joint_inference.py(162) [INFO] - Inference Start
100% | 14042/14042 | 00:04:00:00, 3105.54it/s, Edge=10689, Cloud=3353]
[2025-03-12 06:36:42,974] joint_inference.py(186) [INFO] - Inference Finished
[2025-03-12 06:36:42,975] joint_inference.py(131) [INFO] - Release models
[2025-03-12 06:36:49,318] hard_sample_mining.py(253) [INFO] - OracleRouter Statistics:
Both Wrong: 2496, Both Correct: 7317, Edge Better: 876, Cloud Better: 3353
[2025-03-12 06:36:51,683] edge_model.py(43) [INFO] - {'model': 'Qwen/Qwen2.5-3B-Instruct', 'backend': 'vllm', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.05,
'tensor_parallel_size': 4, 'gpu_memory_utilization': 0.9, 'use_cache': True}
[2025-03-12 06:36:51,683] cloud_model.py(34) [INFO] - {'model': 'gpt-4o-mini', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.05, 'use_cache': True}
[2025-03-12 06:36:51,692] joint_inference.py(73) [INFO] - Loading dataset
[2025-03-12 06:36:52,797] hard_sample_mining.py(30) [INFO] - USING OracleRouterFilter
[2025-03-12 06:36:52,798] joint_inference.py(162) [INFO] - Inference Start
100% | 14042/14042 | 00:04:00:00, 3191.63it/s, Edge=10889, Cloud=3153]
[2025-03-12 06:36:57,198] joint_inference.py(186) [INFO] - Inference Finished
[2025-03-12 06:36:57,198] joint_inference.py(131) [INFO] - Release models
[2025-03-12 06:37:03,005] hard_sample_mining.py(253) [INFO] - OracleRouter Statistics:
Both Wrong: 2422, Both Correct: 7517, Edge Better: 950, Cloud Better: 3153
[2025-03-12 06:37:05,167] edge_model.py(43) [INFO] - {'model': 'Qwen/Qwen2.5-7B-Instruct', 'backend': 'vllm', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.05,
'tensor_parallel_size': 4, 'gpu_memory_utilization': 0.9, 'use_cache': True}
[2025-03-12 06:37:05,167] cloud_model.py(34) [INFO] - {'model': 'gpt-4o-mini', 'temperature': 0, 'top_p': 0.8, 'max_tokens': 512, 'repetition_penalty': 1.05, 'use_cache': True}
[2025-03-12 06:37:05,176] joint_inference.py(73) [INFO] - Loading dataset
[2025-03-12 06:37:06,423] hard_sample_mining.py(30) [INFO] - USING OracleRouterFilter
[2025-03-12 06:37:06,424] joint_inference.py(162) [INFO] - Inference Start
100% | 14042/14042 | 00:04:00:00, 2937.18it/s, Edge=12304, Cloud=1738]
[2025-03-12 06:37:11,205] joint_inference.py(186) [INFO] - Inference Finished
[2025-03-12 06:37:11,205] joint_inference.py(131) [INFO] - Release models
[2025-03-12 06:37:16,727] hard_sample_mining.py(253) [INFO] - OracleRouter Statistics:
Both Wrong: 2216, Both Correct: 8932, Edge Better: 1156, Cloud Better: 1738
[2025-03-12 06:37:19,271] benchmarking.py(39) [INFO] - benchmarkingjob runs successfully.
```

rank	algorithm	Accuracy	Edge Ratio	Time to First Token	Throughput	Internal Token Latency	Cloud Prompt Tokens	Cloud Completion Tokens	Edge Prompt Tokens	Edge Compl
							time			url
1	query-routing	84.22	87.62	0.347	170.28	0.006	1560307	20339	10695142	30
184	jointinference	OracleRouter	Qwen/Qwen2.5-7B-Instruct	vllm	gpt-4o-mini	2024-10-28 16:58:30	./workspace-mmLu/benchmarkingjob/query-routi			
2	query-routing	82.75	77.55	0.316	216.72	0.005	2727792	18177	9470276	29
1364	jointinference	OracleRouter	Qwen/Qwen2.5-3B-Instruct	vllm	gpt-4o-mini	2024-10-28 16:58:19	./workspace-mmLu/benchmarkingjob/query-routi			
126	jointinference	OracleRouter	Qwen/Qwen2.5-1.5B-Instruct	vllm	gpt-4o-mini	2024-10-28 16:58:09	./workspace-mmLu/benchmarkingjob/query-routi			
0	query-routing	75.99	0.0	0.691	698.83	0.001	11739216	79115	0	
	jointinference	CloudOnly	Qwen/Qwen2.5-1.5B-Instruct	vllm	gpt-4o-mini	2024-10-28 16:57:43	./workspace-mmLu/benchmarkingjob/query-routi			




Part 04

开源成果分享

AI





[Python](#)
[Pytorch](#)
[KubeEdge-Ianvs](#)
[LLM](#)

CNCF - KubeEdge: Cloud-Edge Speculative Decoding for LLM via KubeEdge-Ianvs (2024 Term 3)

[View Repository](#)
[View Site](#)

[upsource](#)
[list](#)
[practice](#)
[history](#)

[Terms](#)

[Term 3: Sep-Nov](#)


The autoregressive decoding mode of LLM determines that LLM can only be decoded serially, which limits its inference speed. Speculative decoding technique can be used to decode LLM in parallel with the help of draft model, so as to improve the inference speed of LLM without loss of accuracy. However, the speculative decoding technology of LLM does not consider the application in the cloud-edge distributed environment. This project aims to implement cloud-edge collaborative speculative decoding based on KubeEdge-Ianvs, an open source cloud-edge collaborative distributed machine learning platform, so...

[View More](#)



[Code of Conduct](#)

Applications Closed



Mentees



Mentors

Sponsor Organizations

[illegible]

<div> <div>最具潜力奖</div> <div>OSPP2024</div> <div></div> </div>			
学生姓名	项目名称	社区名称	社区导师
范琦	大语言模型云边协同推理：基于KubeEdge-lanvs实现	KubeEdge	胡时京

<https://github.com/kubeedge/ianvs/tree/main/examples/cloud-edge-collaborative-inference-for-llm>

KubeEdge SIG AI：完善分布式协同AI应用生态



分布式协同
AI框架

分布式协
同AI基准
测试

KubeEdge
SIG AI

分布式协同AI框架（Sedna）

定义AI应用分布式化的编程框架，帮助开发者快速开发边云协同AI应用，使AI应用更好地在边缘运行（包括成本节约、性能提升和数据保护）。

SIG AI已经孵化边云协同AI框架Sedna子项目，并在AI领域形成一定影响。

分布式协同AI基准测试（Ianvs）

在开发、评估分布式协同AI应用和服务系统时，帮助用户确定关键维度性能：

- 分布式协同AI关键特性的全面基准规格；
- 分布式协同AI典型场景的测试用例；
- 分布式协同AI端到端测试床；

欢迎加入我们: kubedge.slack.com

● Kubeedge SIG AI

➤ Github: <https://github.com/kubedge/community/tree/master/sig-ai>



Kubeedge社区公众号



添加社区小助手微信，
发送 SIG AI 讲群



演讲者胡时京微信，输
入验证信息“KCD北
京”



演讲者范或微信，输入
验证信息“KCD北京”

Thanks.

AI

