



CoCoMaMa: Contextual Combinatorial Multi-Armed Bandit Router for Multi-Agent Systems with Volatile Arms

Motivation | Architecture | Related Work | CoCoMaMa Approach | Evaluation | Outlook

Jonathan Rau, Jonathan Bader, Philipp Wiesner, Odej Kao | Distributed and Operating Systems (DOS) Group | TU Berlin



1. MOTIVATION

THE WEB-AGENT LANDSCAPE

```
{
  "name": "GeoSpatial Route Planner Agent",
  "description": "[...]",
  "url":
  "https://georoute-agent.example.com/a2a/v1",
  "version": "1.2.0",
  "capabilities": {
    "streaming": true,
    "pushNotifications": true,
    "stateTransitionHistory": false
  },
  "securitySchemes": {...},
  "skills": [{
    "id": "route-optimizer-traffic",
    "name": "Traffic-Aware Route Optimizer",
    "description": "[...]",
  }]
}
```

Diversity in

- Architectures: BDI, Reactive, LLM-based
- Techstacks: Langgraph, ReACT
- Available Sensors: Visual, Audio, GPS
- Available Affordances: bound to a robot, operating a web browser, access to protected IT-Systems via MCP
- Goals: transport goods, customer support

Interoperability between Agents

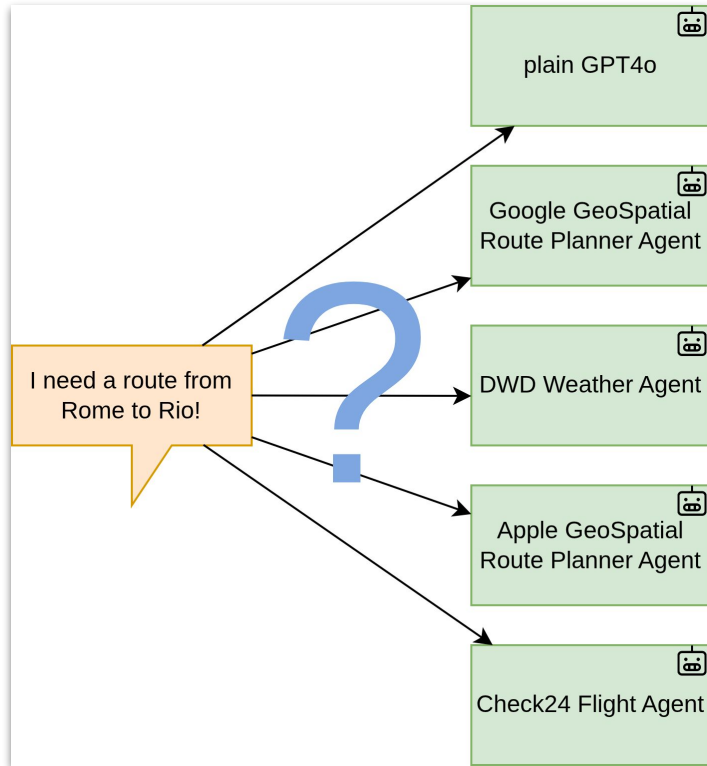
- Agent2Agent Protocol
- Agent Network Protocol
- Eclipse LMOS

Redundancy due to

- Competing actors in open markets
- Slightly different versions of the same agent

1. MOTIVATION

THE ROUTING PROBLEM



Assumptions

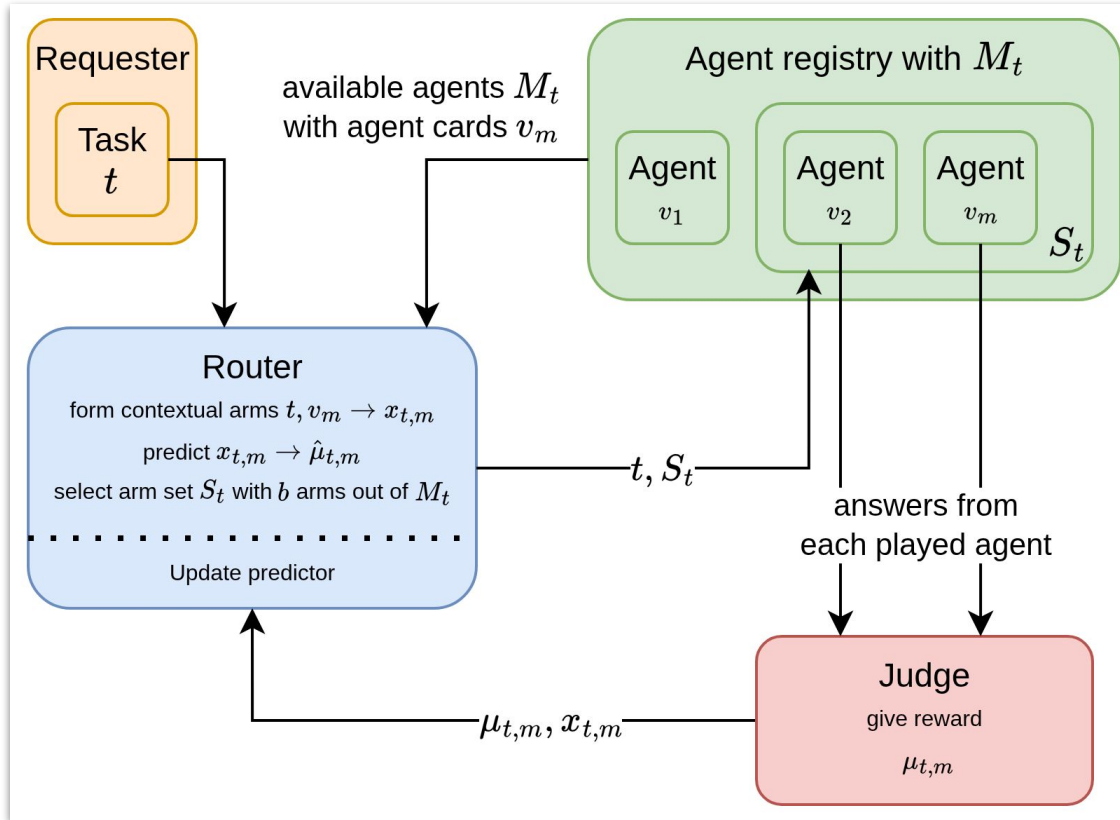
- Individual experts outperform general purpose agents in their respective expert domain
- Performance depends on the **context** of the task
- Available agents are **volatile**
- A single task can be assigned to a **combination** of agents
- Performance is observed after assigning a task to an agent
- **No prior knowledge** from benchmarking datasets

Goals

- Explore the available options and learn from feedback
- Exploit the best performing agents for each task

2. ARCHITECTURE

A MULTI AGENT SYSTEM WITH A FEEDBACK-LOOP



Assumptions

- Judge is given and gives “true” rewards between 0,1
- Budget b represents the amount of arms to play per round
- Total amount of rounds t not known in advance

3. RELATED WORK

LLM Model Routing: Offline learning with a fixed number of models

- Binary classifier for each model (Shnitzer et. al.)
- Pairwise comparison using random forests (RoRF)
- Train a task encoder and model embeddings (RouterDC)
- Train KNN on task embeddings or finetune RoBERTa (CARROT)

LLM Model Routing: Online Learning with a fixed number of models

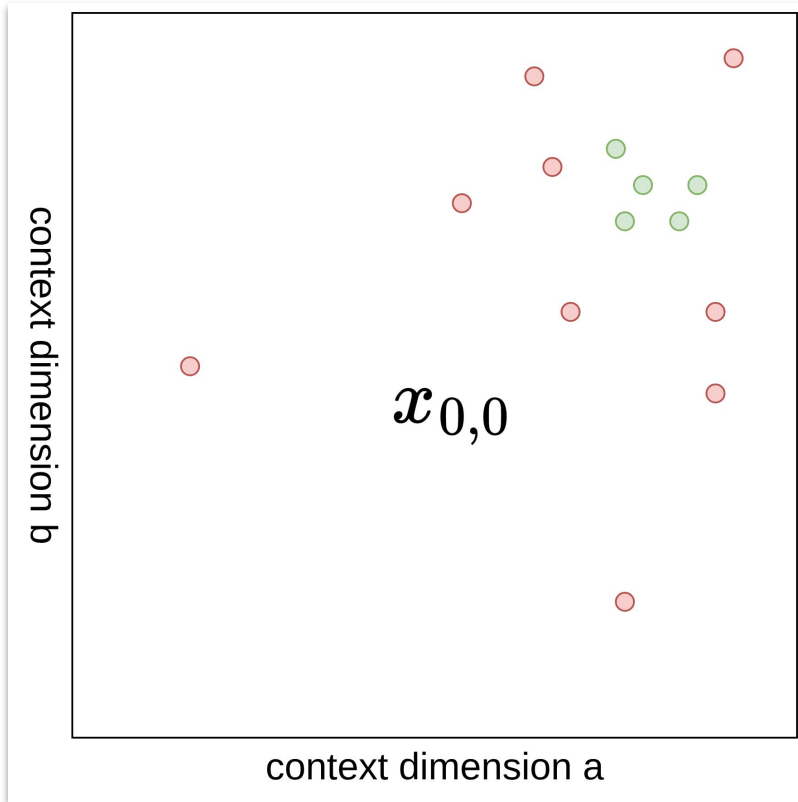
- Gradient ascent and ϵ -greedy Q-Learning (PickLLM)
- Pretrain in offline phase, update MLP in online phase (MixLLM)

Contextual, Combinatorial Multi-Armed Bandits with volatile arms

- Multi-Armed Bandits: a player repeatedly chooses from multiple options (arms) with unknown reward distributions, balancing exploration and exploitation
- Contextual Bandits: additional context information is available to tailor action choices for better rewards
- Combinatorial Bandits: a combination of arms is chosen, optimizing joint rewards
- Volatile arms: the set of available arms can change over time

3. RELATED WORK

ACC-UCB ALGORITHM

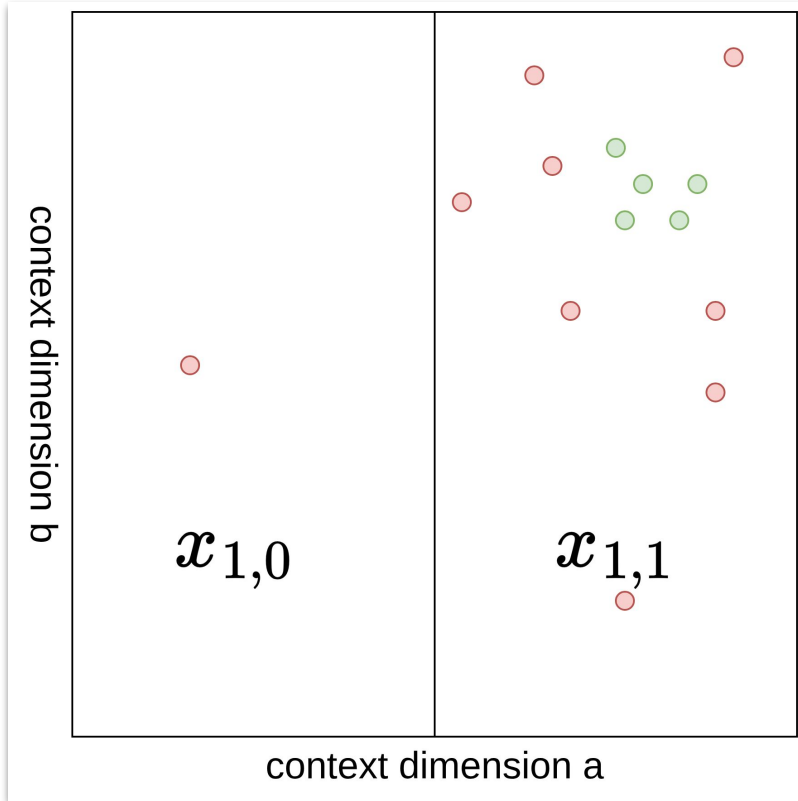


Adaptive Discretization Upper Confidence Bound by Nika et al.

- Split the context space into non-overlapping regions using a tree based approach
- A region is defined as a set of hypercubes
 - We propose HD-ACC-UCB using hyperrectangles for reduced memory consumption instead
- Estimate the reward based on average performance of the region
- Balance node selection on highest expected reward (exploit) and confidence (explore)
- Split a region at the center, if the confidence is high

3. RELATED WORK

ACC-UCB ALGORITHM

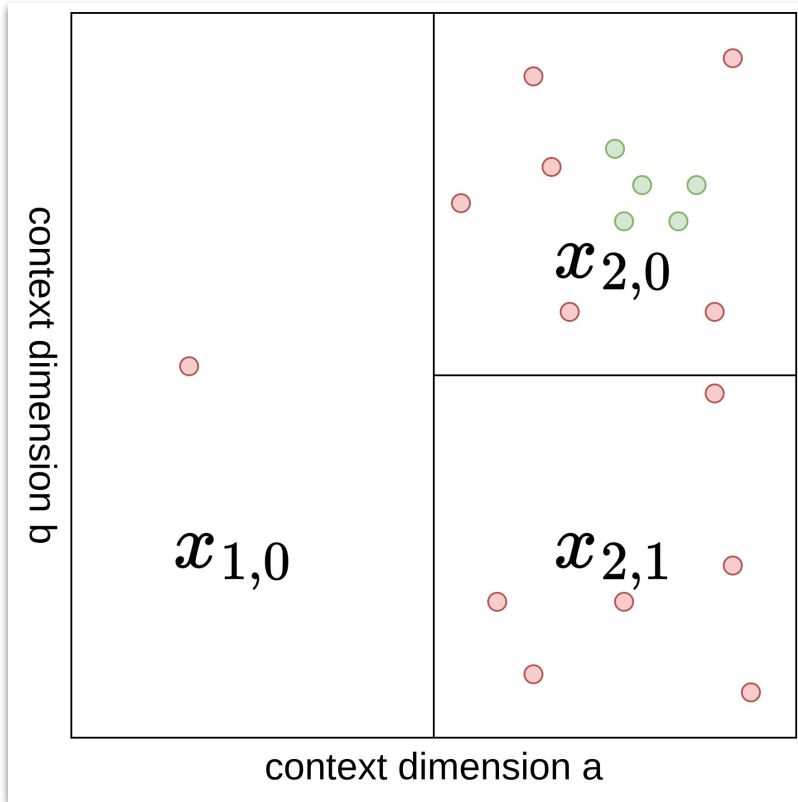


Adaptive Discretization Upper Confidence Bound by Nika et al.

- Split the context space into non-overlapping regions using a tree based approach
- A region is defined as a set of hypercubes
 - We propose HD-ACC-UCB using hyperrectangles for reduced memory consumption instead
- Estimate the reward based on average performance of the region
- Balance node selection on highest expected reward (exploit) and confidence (explore)
- Split a region at the center, if the confidence is high

3. RELATED WORK

ACC-UCB ALGORITHM

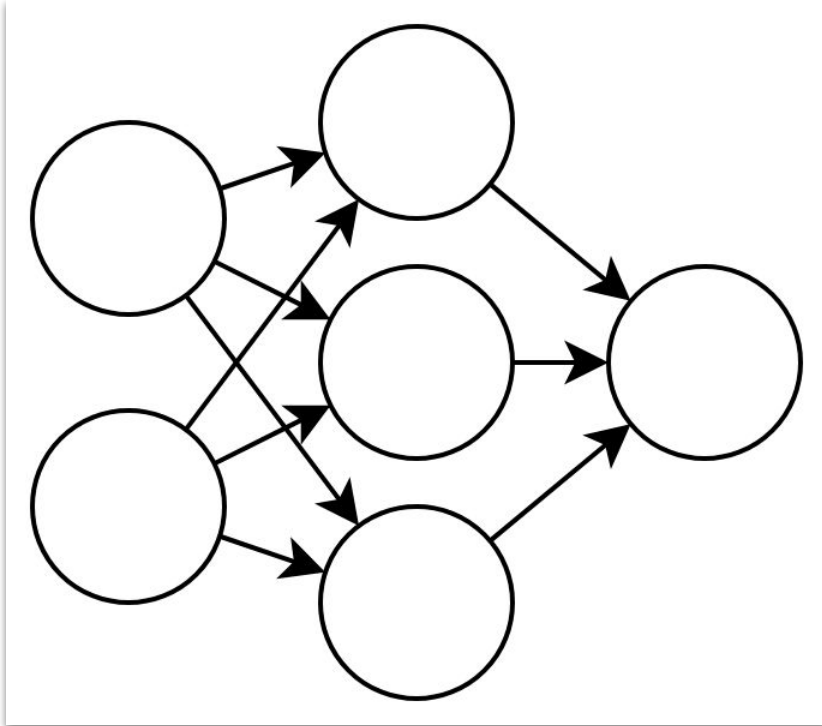


Adaptive Discretization Upper Confidence Bound by Nika et al.

- Split the context space into non-overlapping regions using a tree based approach
- A region is defined as a set of hypercubes
 - We propose HD-ACC-UCB using hyperrectangles for reduced memory consumption instead
- Estimate the reward based on average performance of the region
- Balance node selection on highest expected reward (exploit) and confidence (explore)
- Split a region at the center, if the confidence is high

3. RELATED WORK

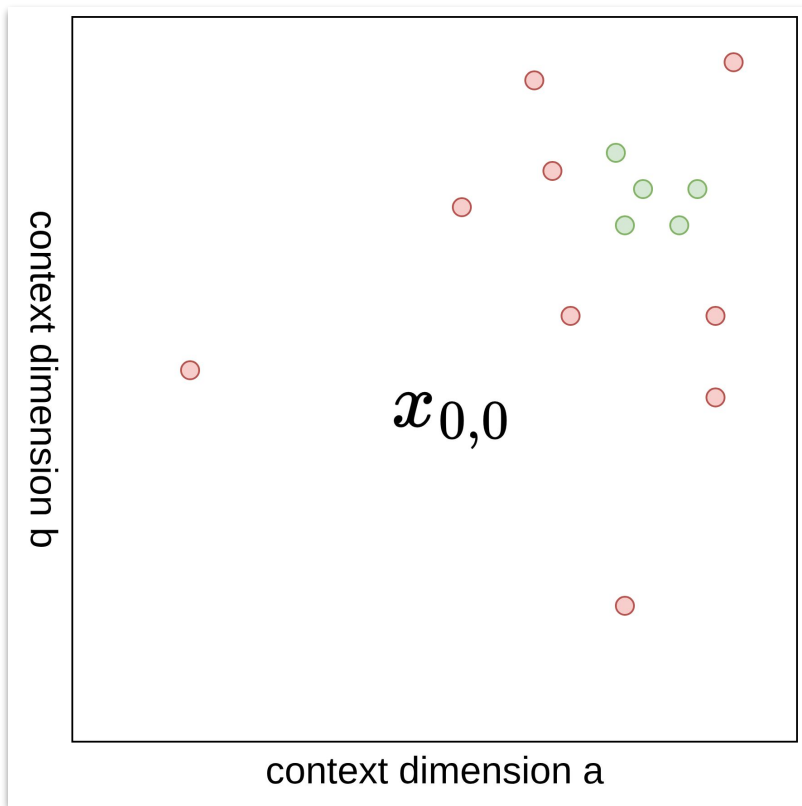
NEURAL MAB ALGORITHM



Neural MAB by Lin et al.

- Use a neural network with 1 hidden layer to predict the reward of each arm
- Use another neural network to find the optimal combination of single rewards
- Greedily select the best agents
- No explicit exploration

4. COCOMAMA APPROACH BASE COCOMAMA



CoCoMaMa

- Split location: at the dimension with the highest covariance, at the mean context value
- Agent selection: based on running mean reward and confidence, incorporating the metrics of the parent node

$$g(x_{h,i}) := \max \begin{cases} \bar{\mu}^t(x_{h,i}) + c^t(x_{h,i}, p(x_{h,i})), \\ \bar{\mu}^t(p(x_{h,i})) + c^t(x_{h,i}, p(x_{h,i})) \end{cases} \quad (1)$$

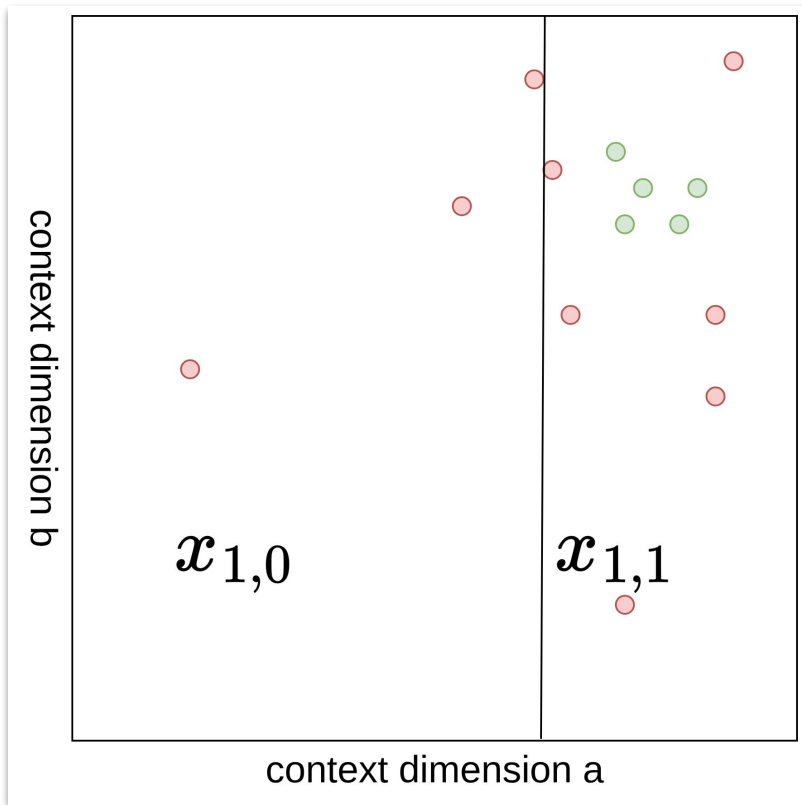
- Split condition: high outcome variance or high confidence, also incorporating the metrics of the parent node

$$\text{Var}^t(\mu(x_{h,i})) > \frac{\theta}{b \cdot t} \cdot \sum_{x \in \mathcal{X}} C^t(x) \cdot \text{Var}^t(\mu(x)) \quad (2)$$

$$C^t(x_{h,i}) > C^t(p(x_{h,i})) \quad (3)$$

$$((2) \text{ and } (3)) \text{ or } c_{\infty}^t(x_{h,i}) \leq v_1 \rho^h.$$

4. COCOMAMA APPROACH BASE COCOMAMA



CoCoMaMa

- Split location: at the dimension with the highest covariance, at the mean context value
- Agent selection: based on running mean reward and confidence, incorporating the metrics of the parent node

$$g(x_{h,i}) := \max \begin{cases} \bar{\mu}^t(x_{h,i}) + c^t(x_{h,i}, p(x_{h,i})), \\ \bar{\mu}^t(p(x_{h,i})) + c^t(x_{h,i}, p(x_{h,i})) \end{cases} \quad (1)$$

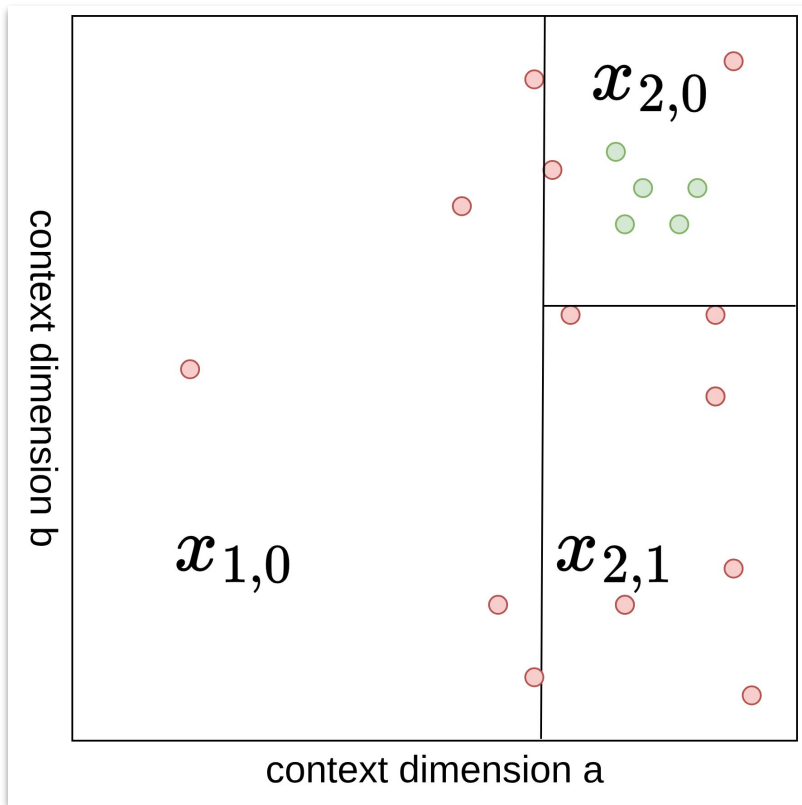
- Split condition: high outcome variance or high confidence, also incorporating the metrics of the parent node

$$\text{Var}^t(\mu(x_{h,i})) > \frac{\theta}{b \cdot t} \cdot \sum_{x \in \dots} C^t(x) \cdot \text{Var}^t(\mu(x)) \quad (2)$$

$$C^t(x_{h,i}) > C^t(p(x_{h,i})) \quad (3)$$

$$((2) \text{ and } (3)) \text{ or } c_{\infty}^t(x_{h,i}) \leq v_1 \rho^h.$$

4. COCOMAMA APPROACH BASE COCOMAMA



CoCoMaMa

- Split location: at the dimension with the highest covariance, at the mean context value
- Agent selection: based on running mean reward and confidence, incorporating the metrics of the parent node

$$g(x_{h,i}) := \max \begin{cases} \bar{\mu}^t(x_{h,i}) + c^t(x_{h,i}, p(x_{h,i})), \\ \bar{\mu}^t(p(x_{h,i})) + c^t(x_{h,i}, p(x_{h,i})) \end{cases} \quad (1)$$

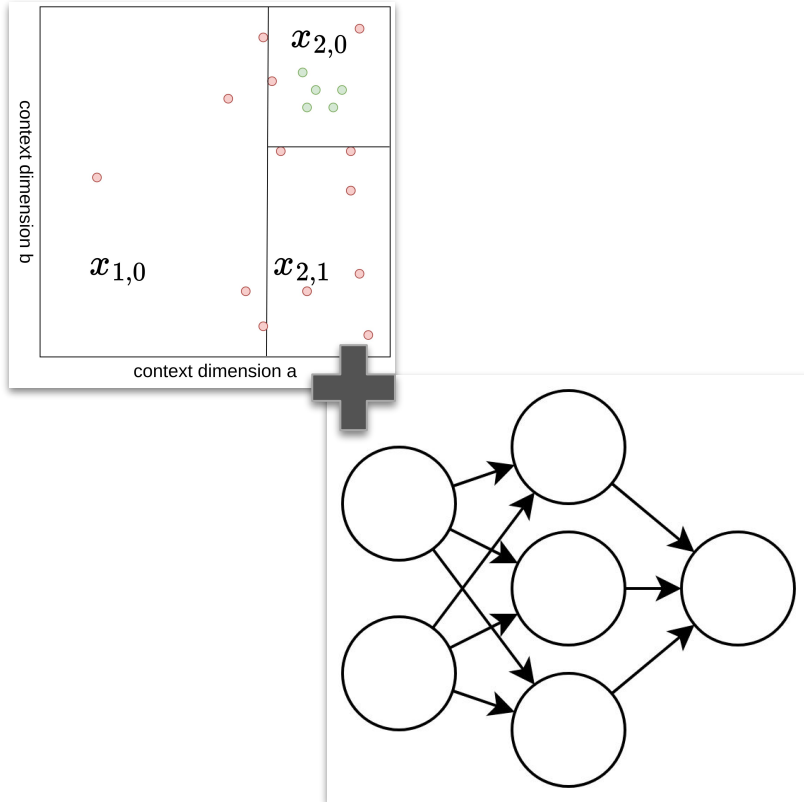
- Split condition: high outcome variance or high confidence, also incorporating the metrics of the parent node

$$\text{Var}^t(\mu(x_{h,i})) > \frac{\theta}{b \cdot t} \cdot \sum_{x \in \dots} C^t(x) \cdot \text{Var}^t(\mu(x)) \quad (2)$$

$$C^t(x_{h,i}) > C^t(p(x_{h,i})) \quad (3)$$

$$((2) \text{ and } (3)) \text{ or } c_{\infty}^t(x_{h,i}) \leq v_1 \rho^h.$$

4. COCOMAMA APPROACH NEURAL COCOMAMA



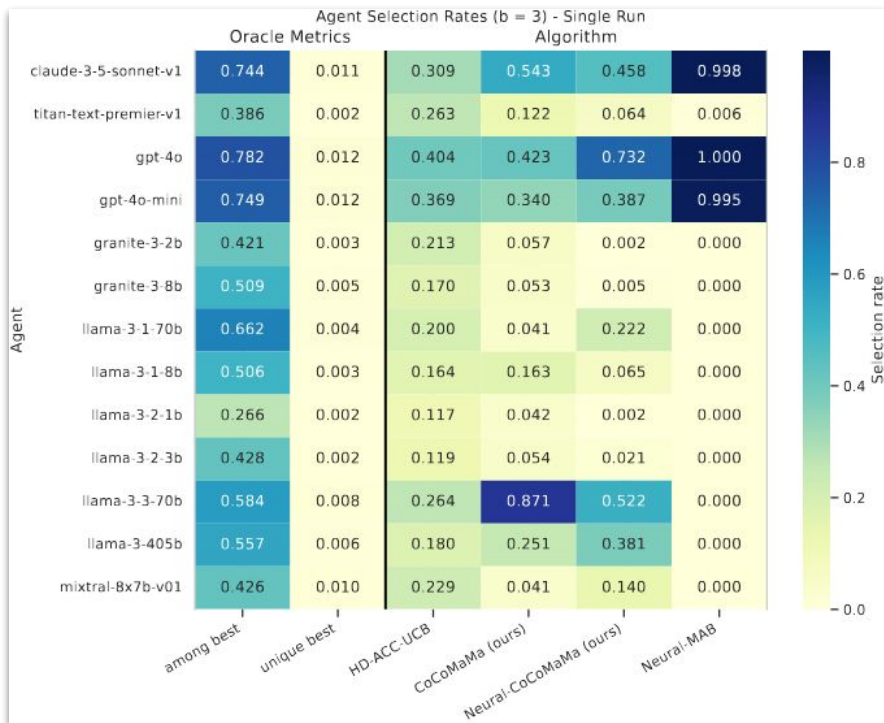
Neural CoCoMaMa

- Split location: same as in base CoCoMaMa
- Agent selection: based on predicted reward and confidence, incorporating the metrics of the parent node

$$g(x_{t,m}) := \hat{\mu}^t(x_{t,m}) + c^t(x_{h,i}, p(x_{h,i})).$$

- Split condition: same as in base CoCoMaMa

5. EVALUATION SPROUT DATASET



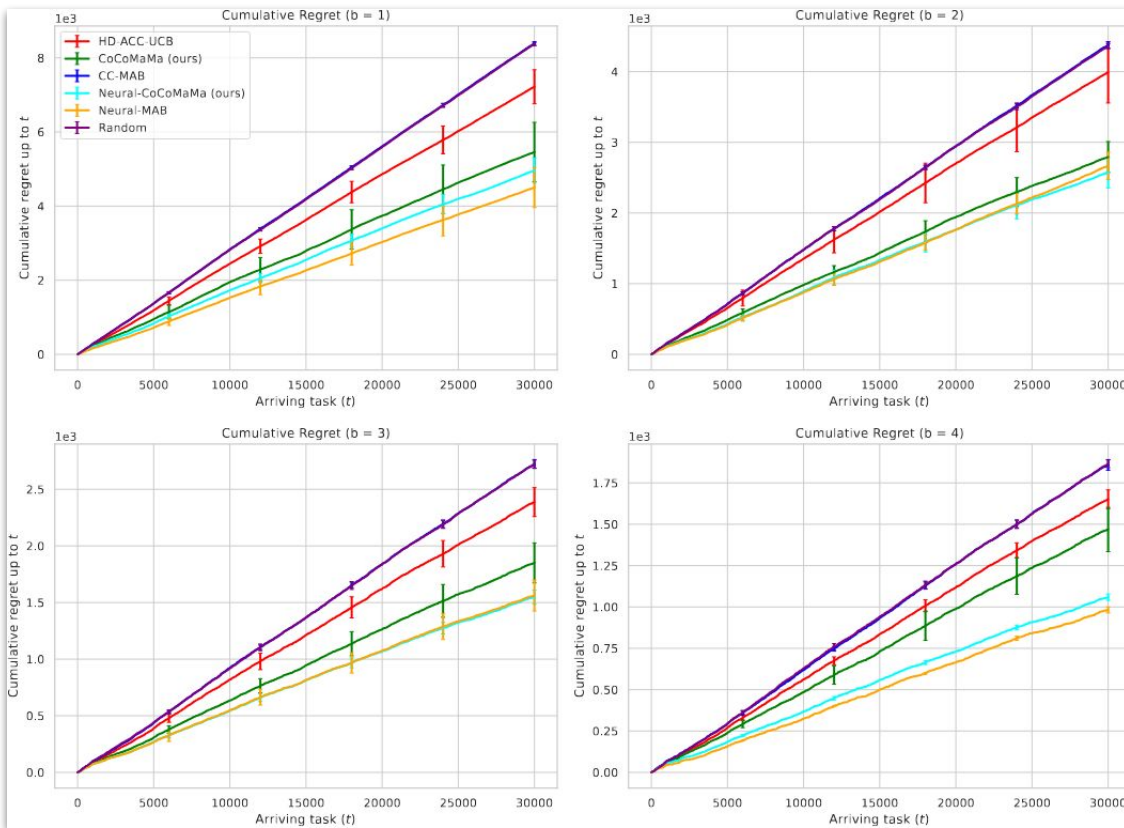
SPROUT by Somerstep et al.

- Queries from 6 popular benchmark datasets (GPQA, MuSR, MMLU-Pro, MATH, OpenHermes, RAGBench)
- Responses from 13 LLMs for each query
- LLaMa-3.1-70b-Instruct is used as a judge to evaluate against ground truth, rating between 0 and 1

Enhanced with A2A agent cards

- generated using perplexity and curated by hand

5. EVALUATION SPROUT DATASET



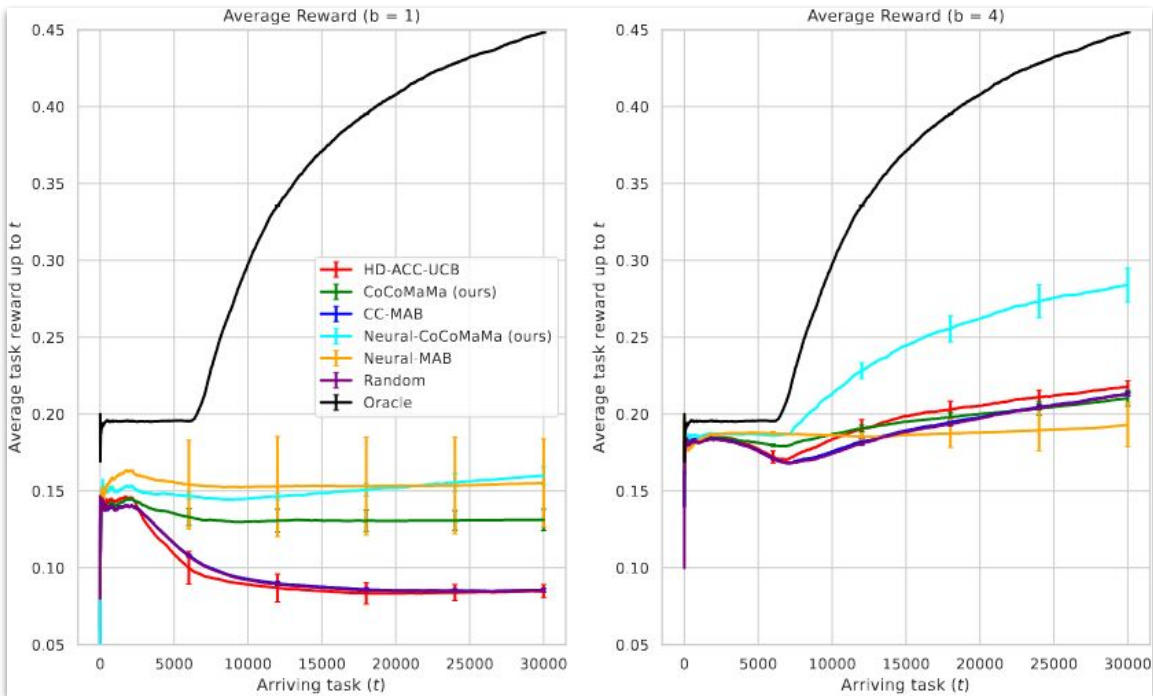
Setup

- tasks and agent cards are embedded using all-MiniLM-L6-v2 with 384 dimensions
- 10 rounds with identical sequential ordering for all budgets

Results

- CoCoMaMa outperforms HD-ACC-UCB for all budgets
- Neural methods outperform methods based on running mean
- Neural-CoCoMaMa matches performance of Neural-MAB for budgets 2 & 3, but adds explainability

5. EVALUATION SYNTHETIC AGENTS DATASET



Setup

- starting with 13 base agents from SPROUT
- weak specialists are added at $t=2000$ every 200 rounds
- strong specialists are added at $t=6000$ every 200 rounds
- score depends 20% on base agent score and 80% on task-agent fit
- base agent score is reduced by 90% if task-agent fit is below threshold

Results

- Neural-MAB fails to explore and exploit the specialized agents
- Over exploring new options hurts the performance
- Neural-CoCoMaMa as only approach to significantly exploit specialists

6. DISCUSSION & FUTURE WORK

Cold start

- Symbolic approaches
- Federated learning
- Better agent cards

Scalability

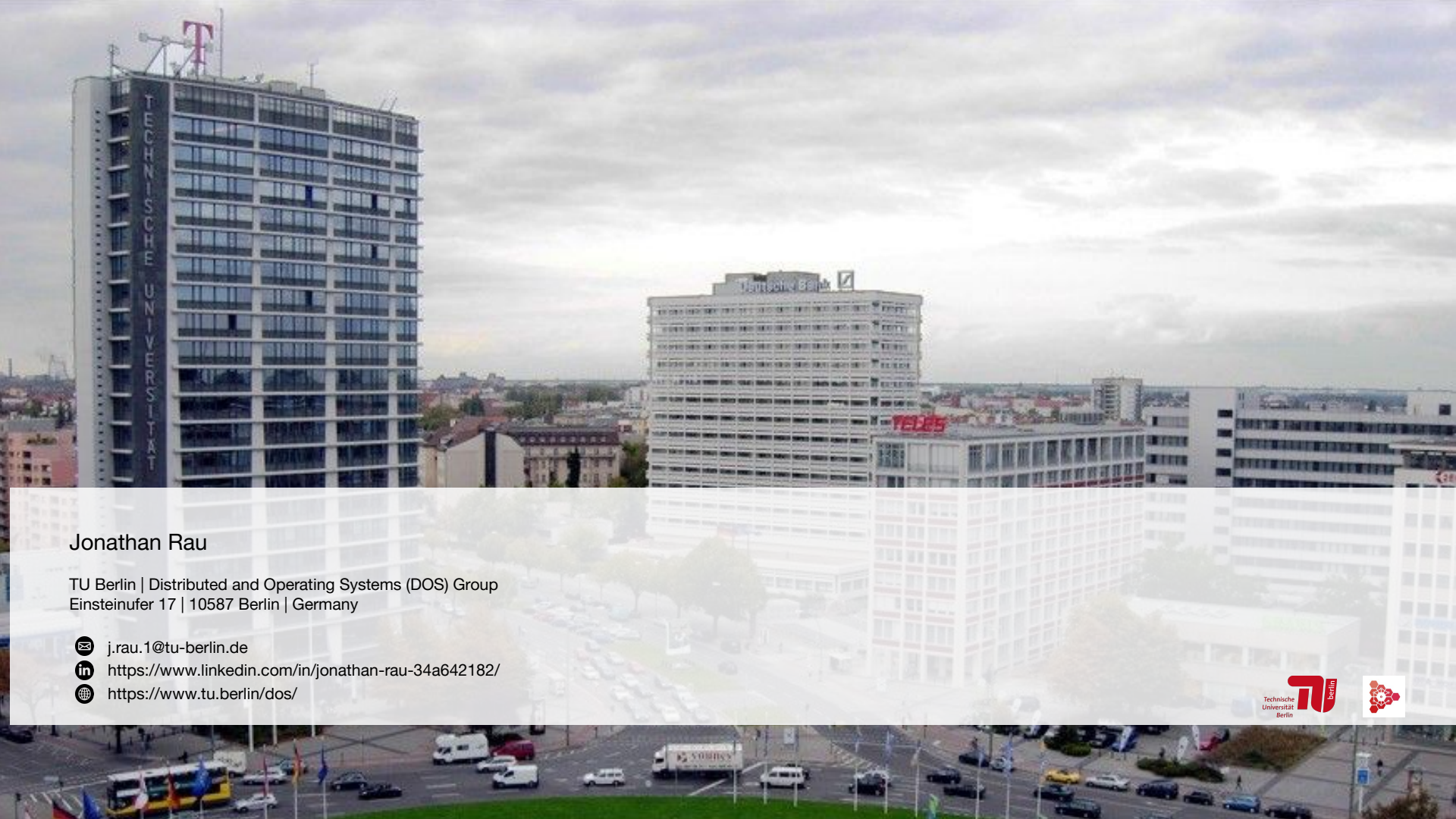
- Retrieval Reranking
- Performance optimizations

Evaluation

- Adversarial robustness?
- Theoretical guarantees and convergence

New Methods

- Distributed Constraint Optimization
- Multi Objective Optimization



Jonathan Rau

TU Berlin | Distributed and Operating Systems (DOS) Group
Einsteinufer 17 | 10587 Berlin | Germany



j.rau.1@tu-berlin.de



<https://www.linkedin.com/in/jonathan-rau-34a642182/>



<https://www.tu.berlin/dos/>

