King Abdullah University of Science and Technology

Università della Svizzera italiana

USI/SUPSI Istituto Dalle Molle di studi sull'intelligenza artificiale IDSIA

Scuola universitaria professionale della Svizzera italiana SUPSI

ICML International Conference On Machine Learning

GPTSwarm: Language Agents as Optimizable Graphs

Mingchen Zhuge, Wenyi Wang, Louis Kirsch[+], Francesco Faccio[+], Dmitrii Khizbullin, Jürgen Schmidhuber[+] ([+]at ICML)

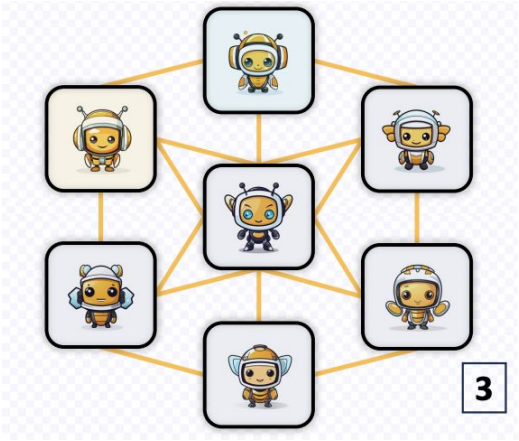
1. Motivation

Current LLM Agents require **significant human engineering** to

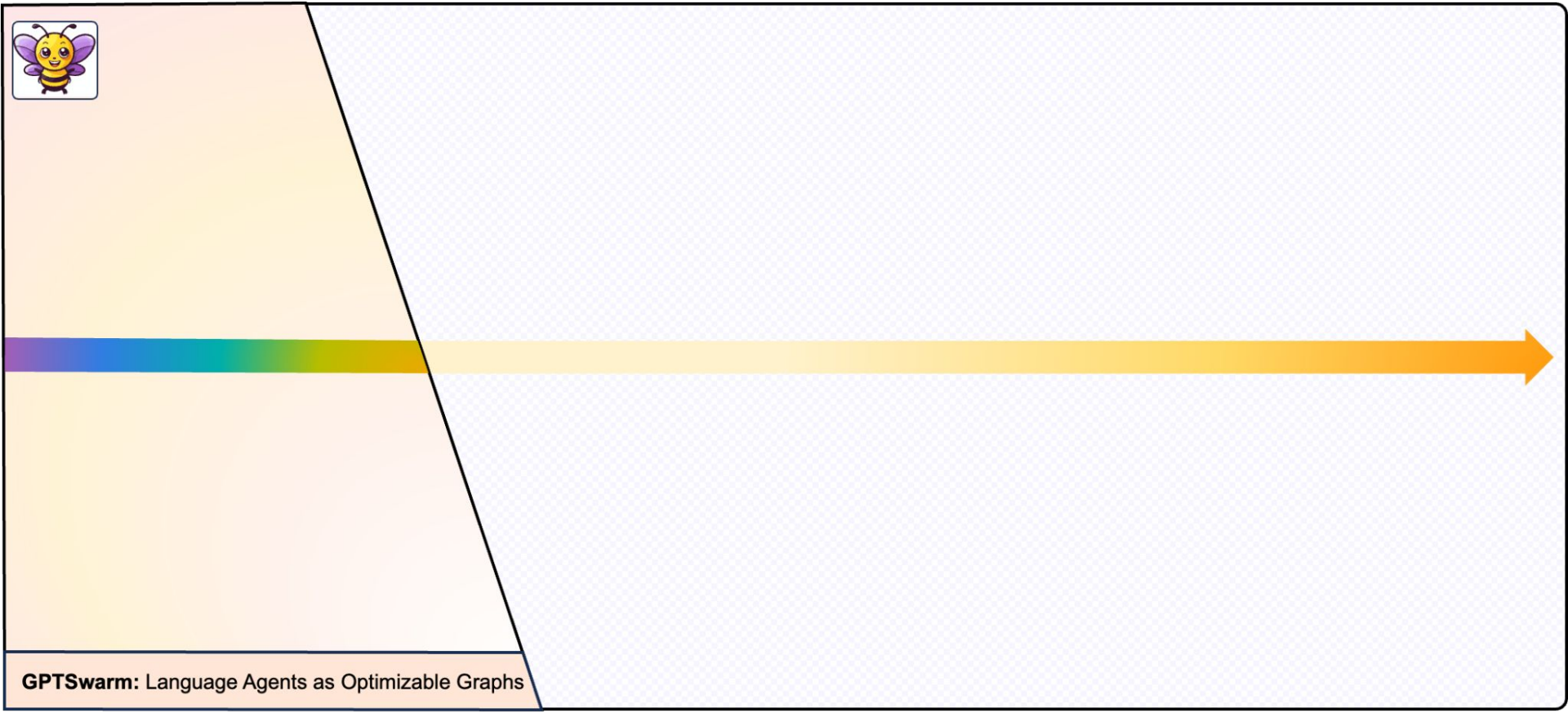
1) Design the inference and orchestration structure
2) Choose appropriate prompts

Can we **unify** different structures and **automatically optimize** over them and the prompts?

Language Agents as Optimizable Graphs



2. Our Approach



2. Our Approach

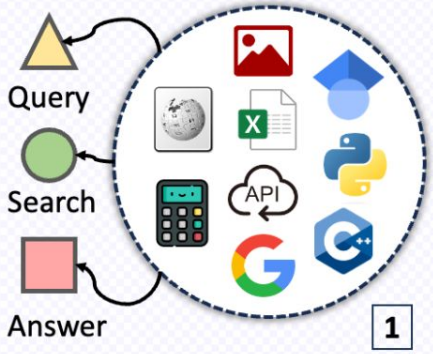
1 Operation ↔ Node

GPTSwarm: Language Agents as Optimizable Graphs

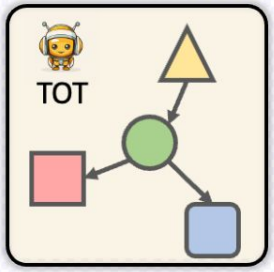
2. Our Approach

1 Operation ↔ Node

2 Agent ↔ Graph of Nodes



1



2

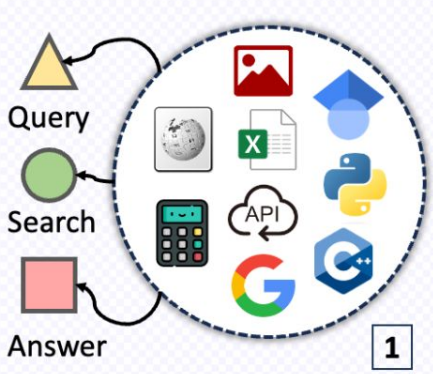
GPTSwarm: Language Agents as Optimizable Graphs

2. Our Approach

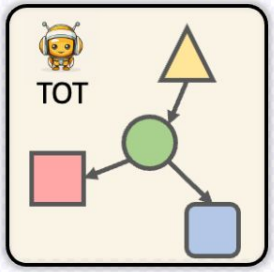
1 Operation ↔ Node

2 Agent ↔ Graph of Nodes

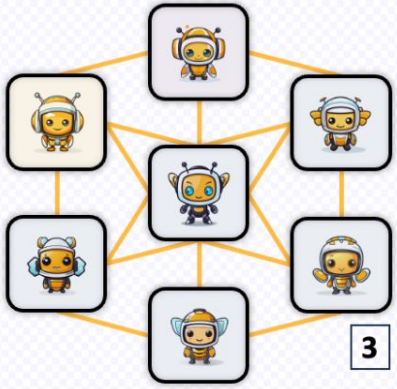
3 Swarm ↔ Composite Graph



1



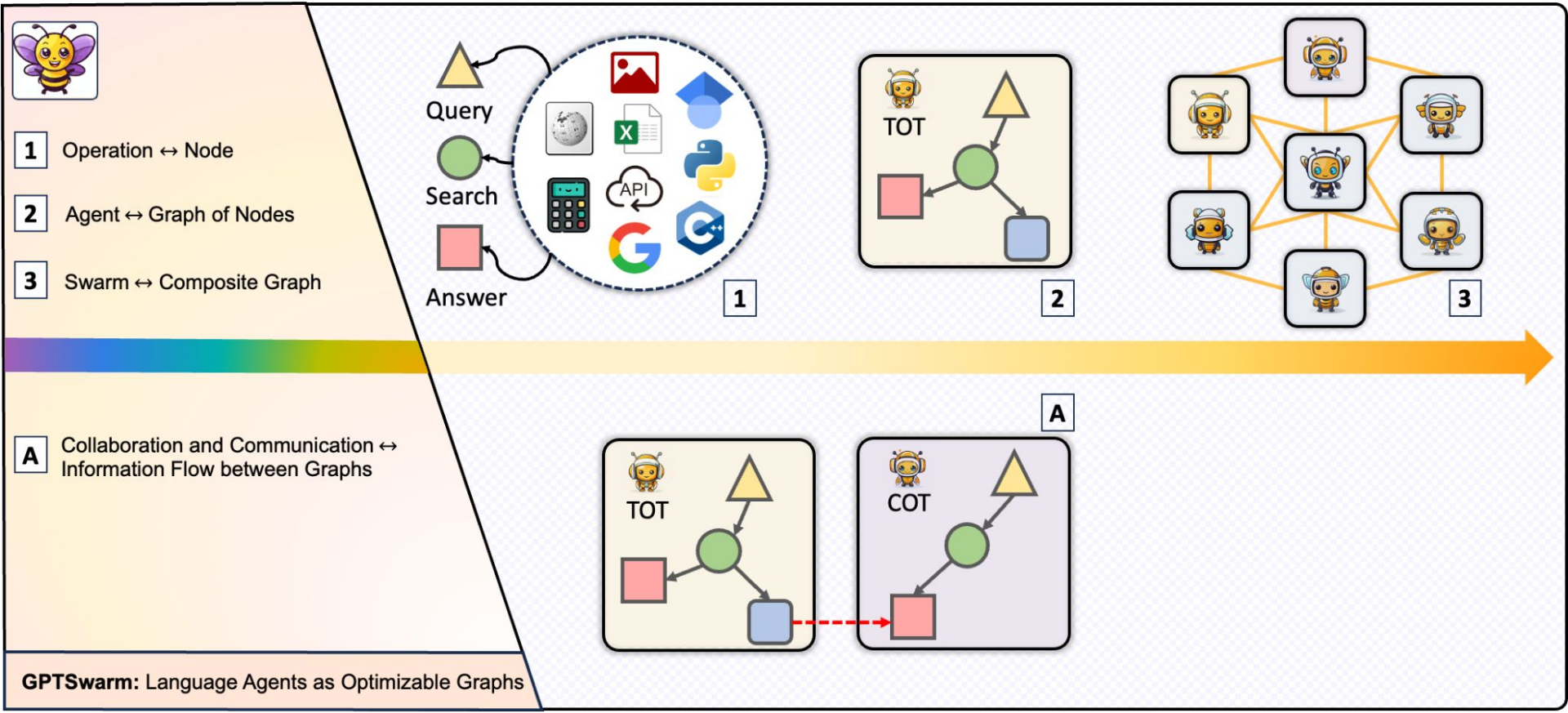
2



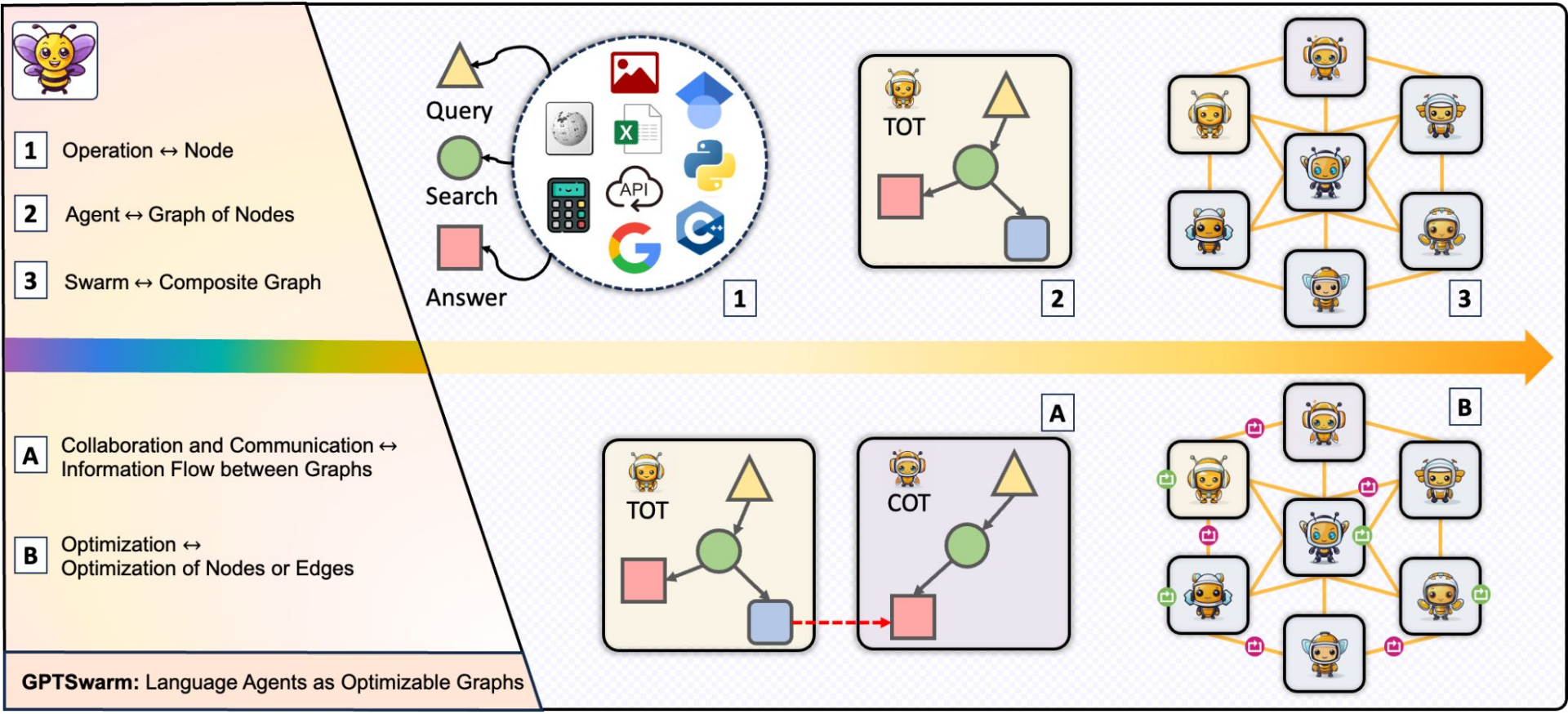
3

GPTSwarm: Language Agents as Optimizable Graphs

2. Our Approach

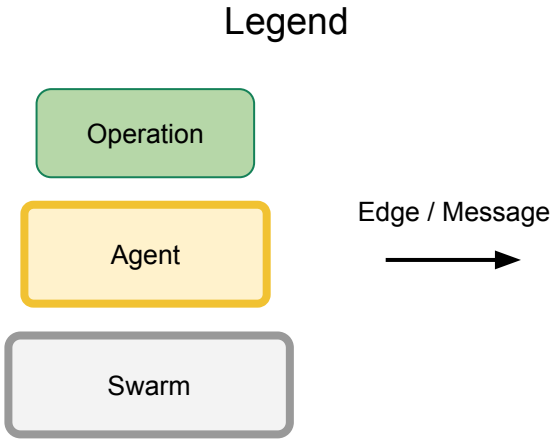


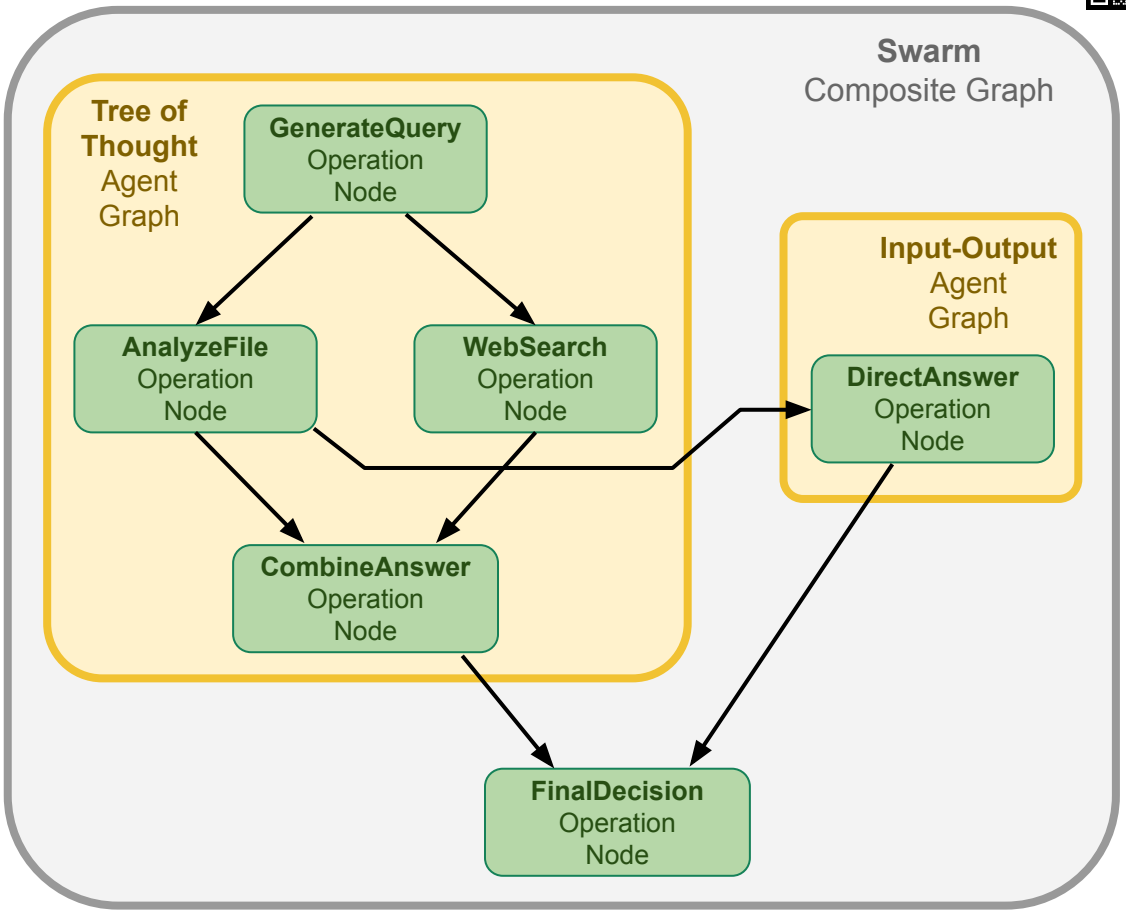
2. Our Approach



3. Algorithms

Composite Graph



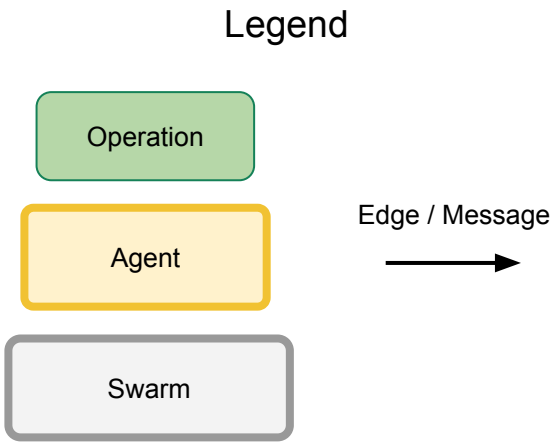


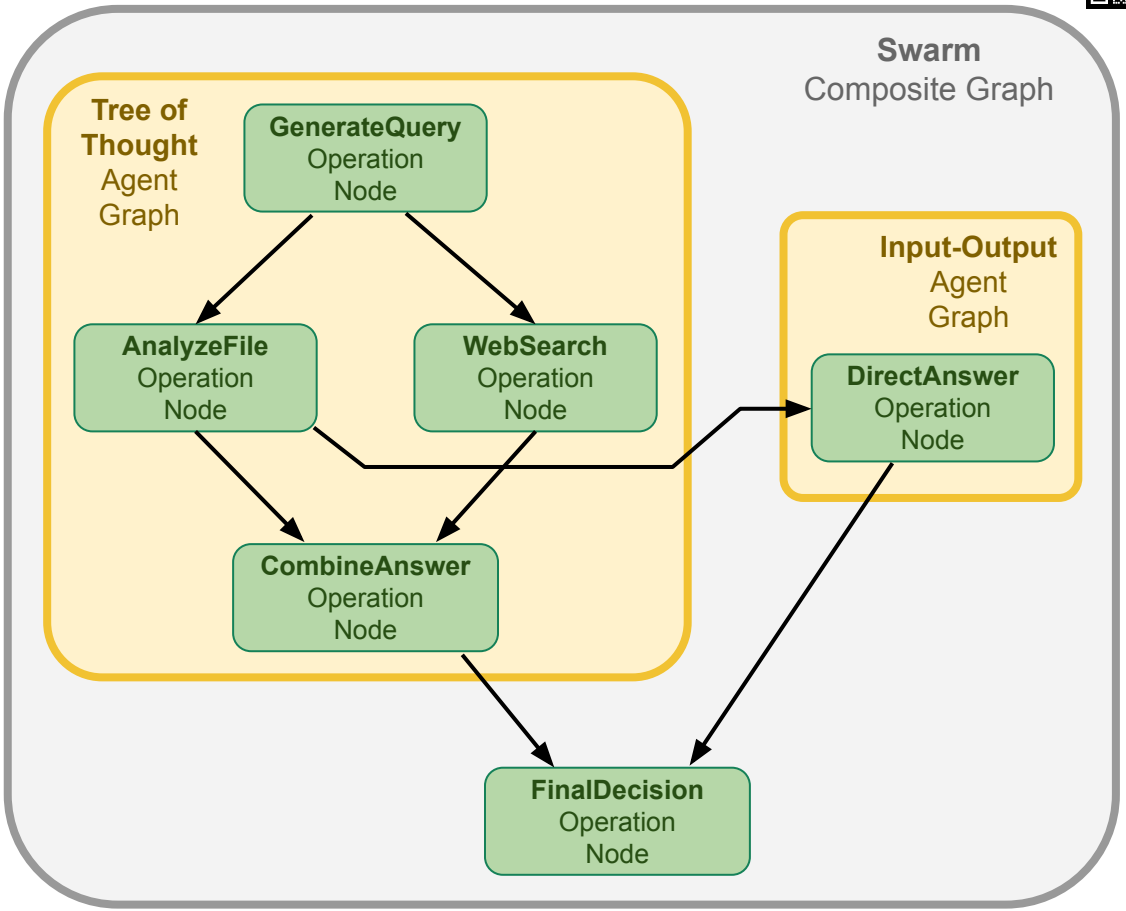
3. Algorithms

Topological Execution

Algorithm 1 Graph Execution

Require: Computational graph $G = (N, E, F, o)$, input x, empty context z for each node without predecessors.
  for n in TopologicalSort(N) **do**
    $z_n \leftarrow \{f_n(z_v, x) : v \in \text{pre}(n)\}$
  end for
Ensure: $f_o(z_o, x)$





3. Algorithms

Edge Optimization

Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ, an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α.
Initialize $\theta \in \mathbb{R}^d$.
while terminate condition not met **do**
  Sample $G_i \sim D_\theta$ for $i = 1, 2, \ldots, M$.
  Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^{M} \hat{u}_\tau(G_i) \nabla_\theta \log(p_\theta(G_i))$.
end while

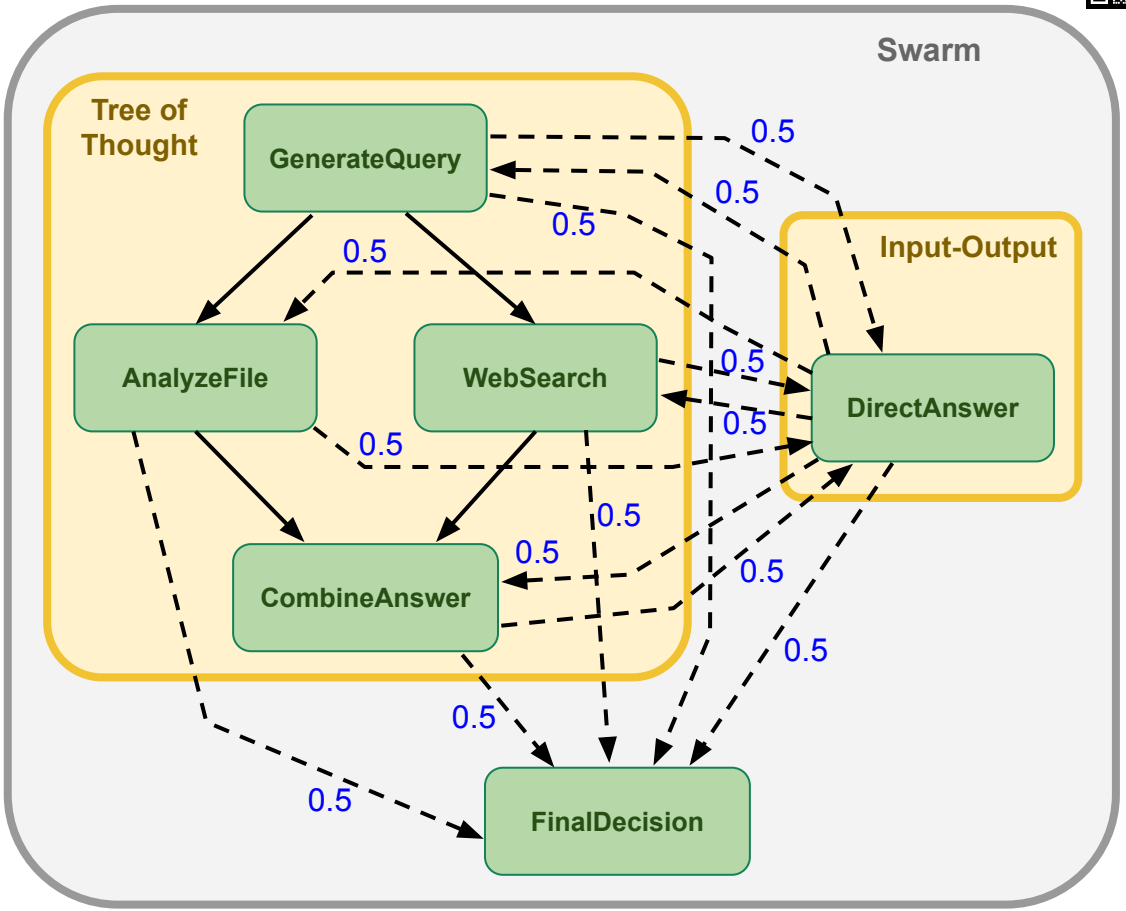
Legend

Fixed Edge

Potential Edge

0.5 - associated probability

1. Initialization



3. Algorithms

2. Random sampling

Edge Optimization

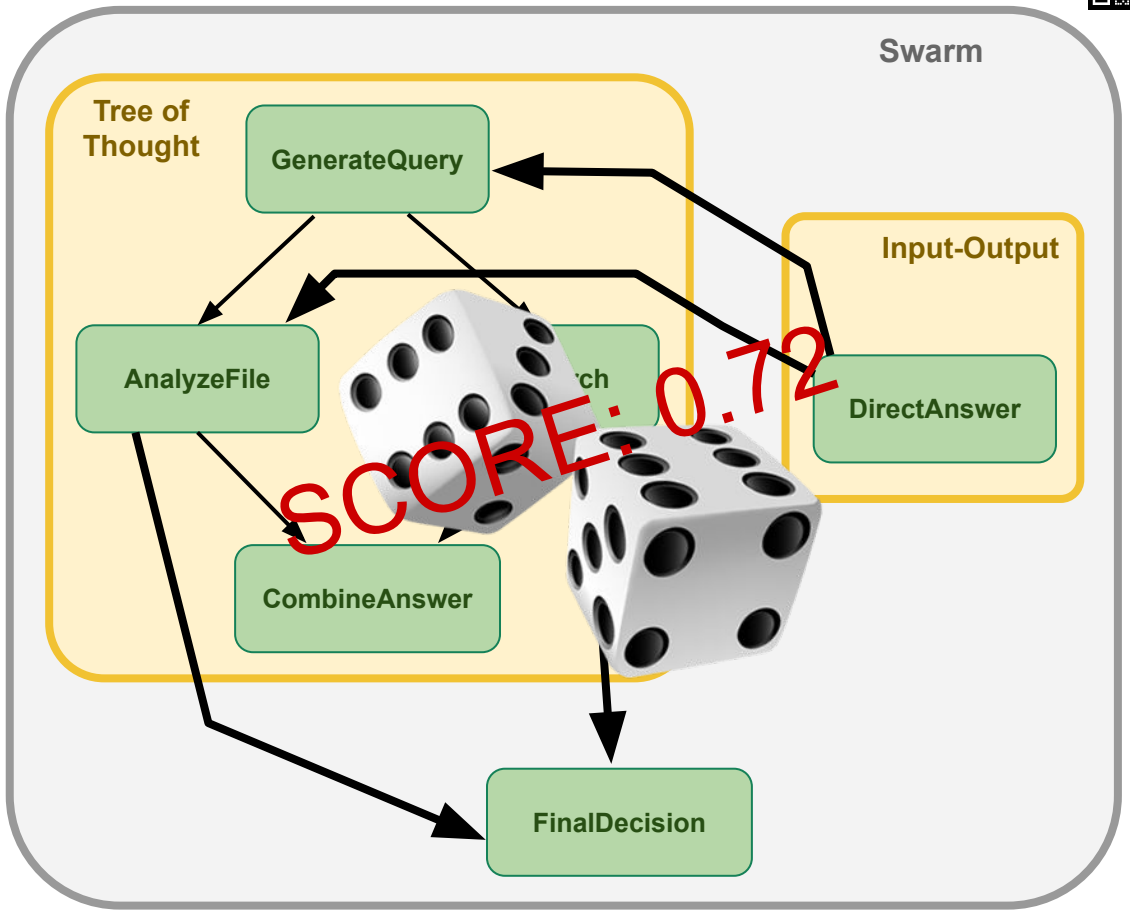
Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ, an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α.
Initialize $\theta \in \mathbb{R}^d$.
while terminate condition not met **do**
  Sample $G_i \sim D_\theta$ for $i = 1, 2, \ldots, M$.
  Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^{M} \hat{u}_\tau(G_i) \nabla_\theta \log(p_\theta(G_i))$.
end while

Legend

Fixed Edge

Realized Edge



3. Algorithms

Edge Optimization

Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ, an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α.
Initialize $\theta \in \mathbb{R}^d$.
while terminate condition not met **do**
  Sample $G_i \sim D_\theta$ for $i = 1, 2, \ldots, M$.
  Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^{M} \hat{u}_\tau(G_i) \nabla_\theta \log(p_\theta(G_i))$.
end while

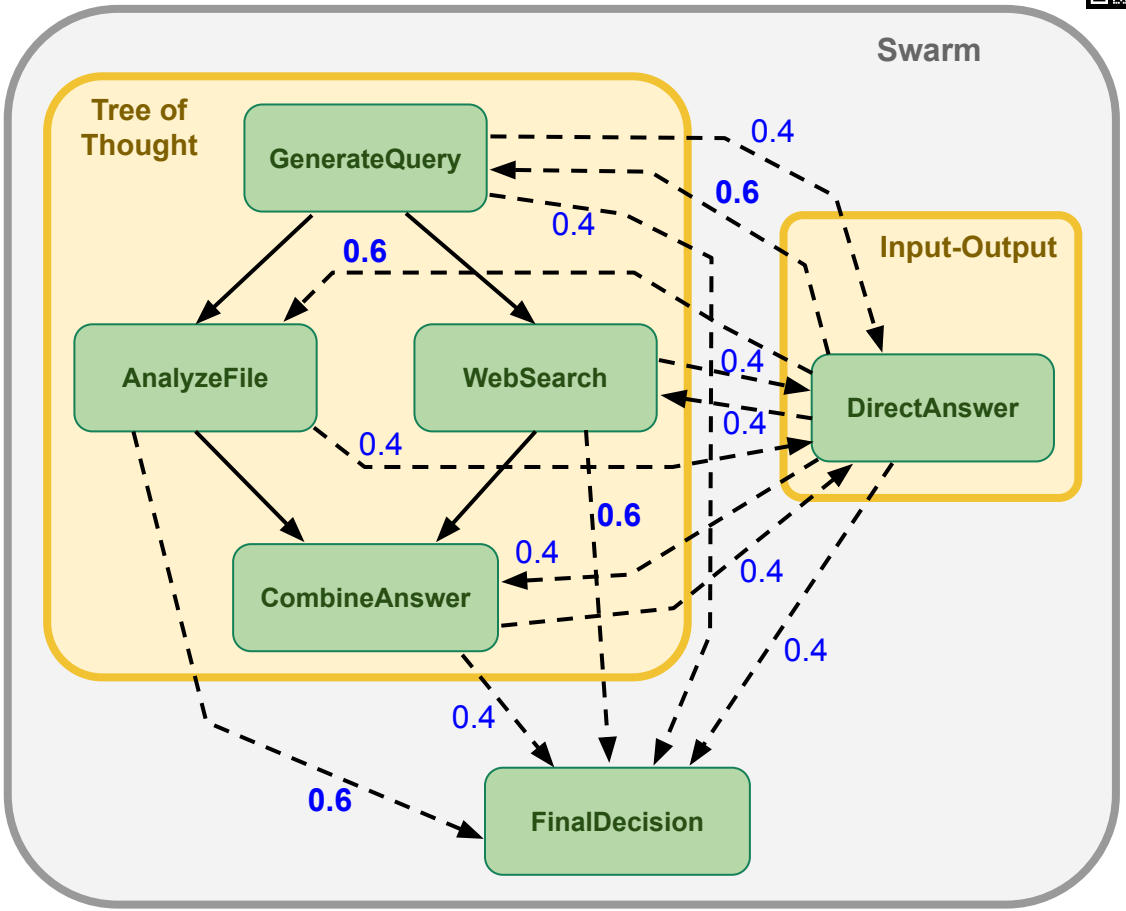
Legend

Fixed Edge

Potential Edge

0.5 - associated probability

3. Update probabilities



3. Algorithms

Edge Optimization

Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ, an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α.
Initialize $\theta \in \mathbb{R}^d$.
while terminate condition not met **do**
  Sample $G_i \sim D_\theta$ for $i = 1, 2, \ldots, M$.
  Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^{M} \hat{u}_\tau(G_i) \nabla_\theta \log(p_\theta(G_i))$.
end while

Legend

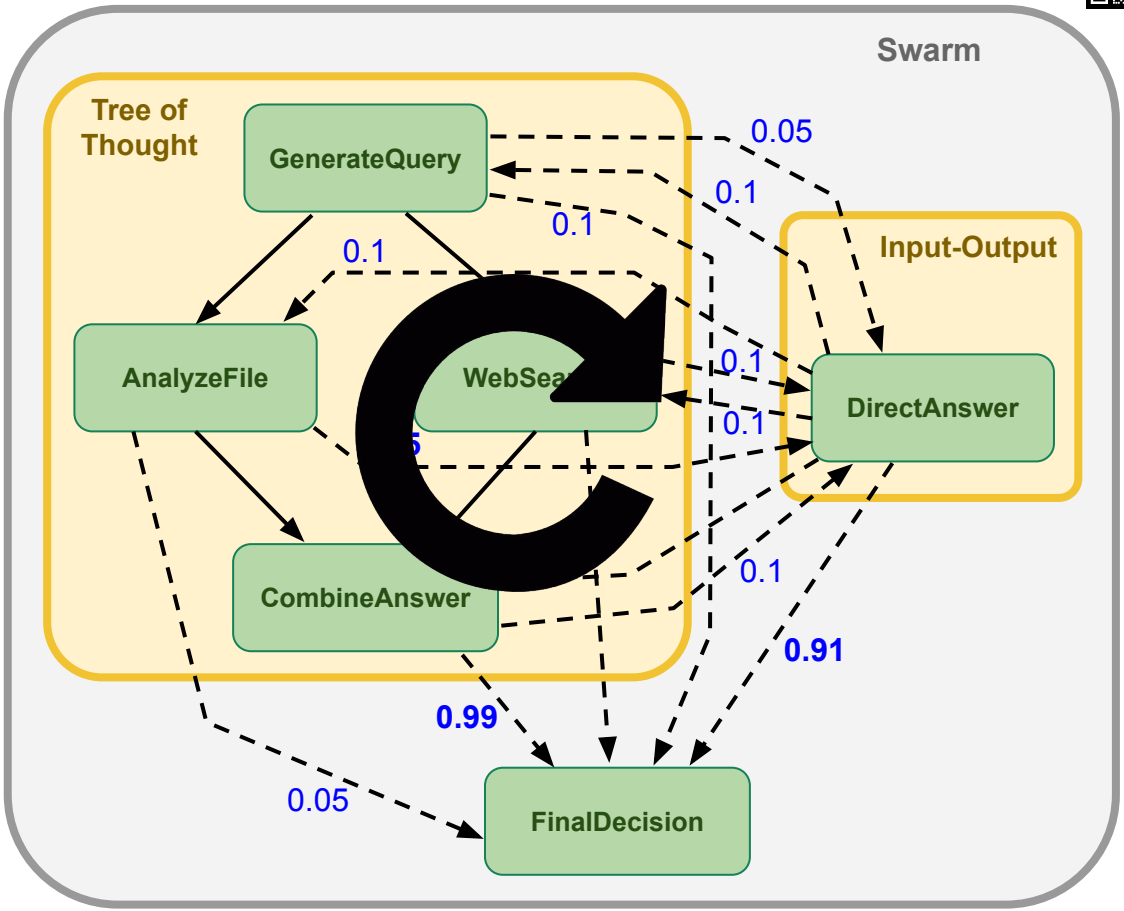
Fixed Edge

Realized Edge

Potential Edge

0.5 - associated probability

4. Final realization



3. Algorithms

Edge Optimization

Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ, an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α.
Initialize $\theta \in \mathbb{R}^d$.
while terminate condition not met **do**
  Sample $G_i \sim D_\theta$ for $i = 1, 2, \ldots, M$.
  Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^{M} \hat{u}_\tau(G_i) \nabla_\theta \log(p_\theta(G_i))$.
end while

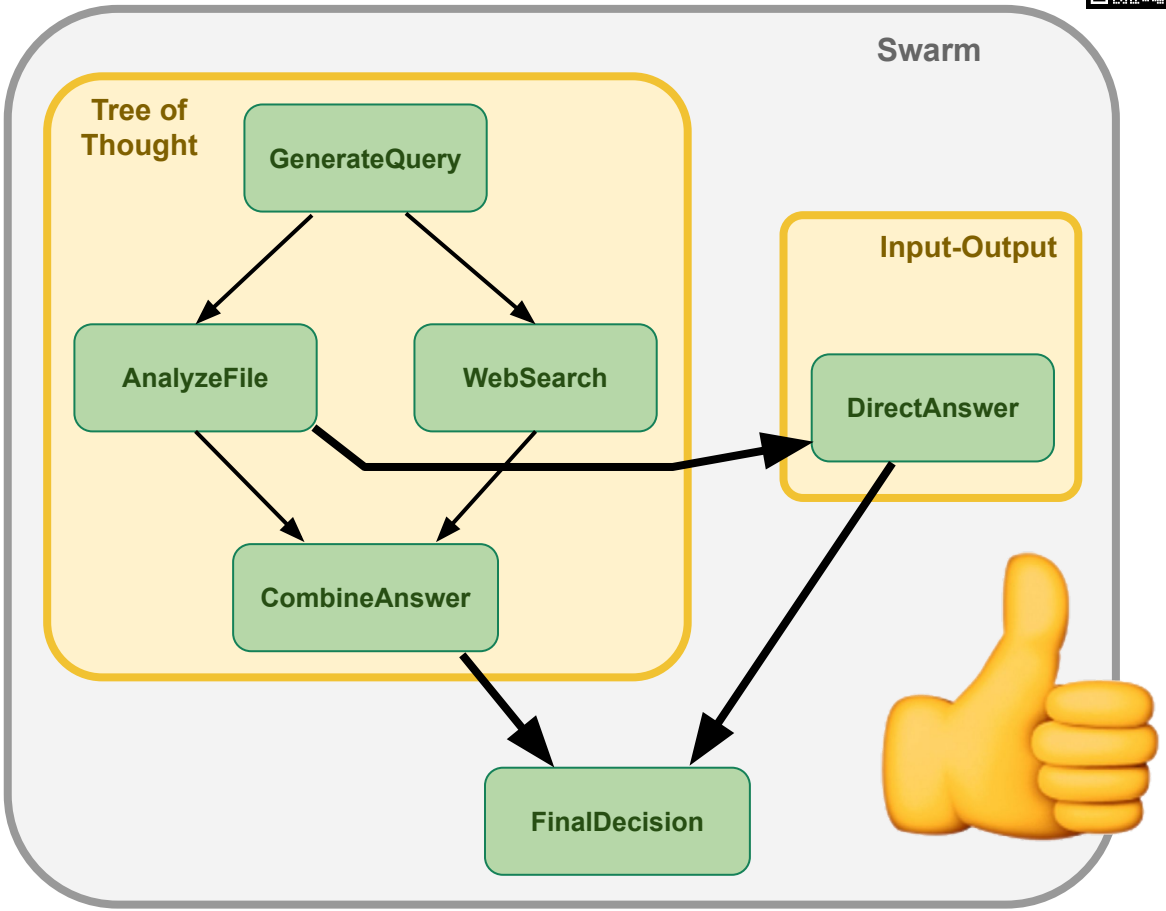
Legend

Fixed Edge

Realized Edge

Potential Edge

4. Final realization



3. Algorithms

Node Optimization

Improve individual operations

Algorithm 3 Node Optimization

Require: A parameterized graph $G^P = (N, E, F^P, o)$, natural language function descriptions $D = \{d_n\}_{n \in N}$, and a distribution of inputs D_X.

Initialize p_n for all $n \in N$.

Initialize $h_n \leftarrow \emptyset$ for all $n \in N$.

while terminate condition not met **do**

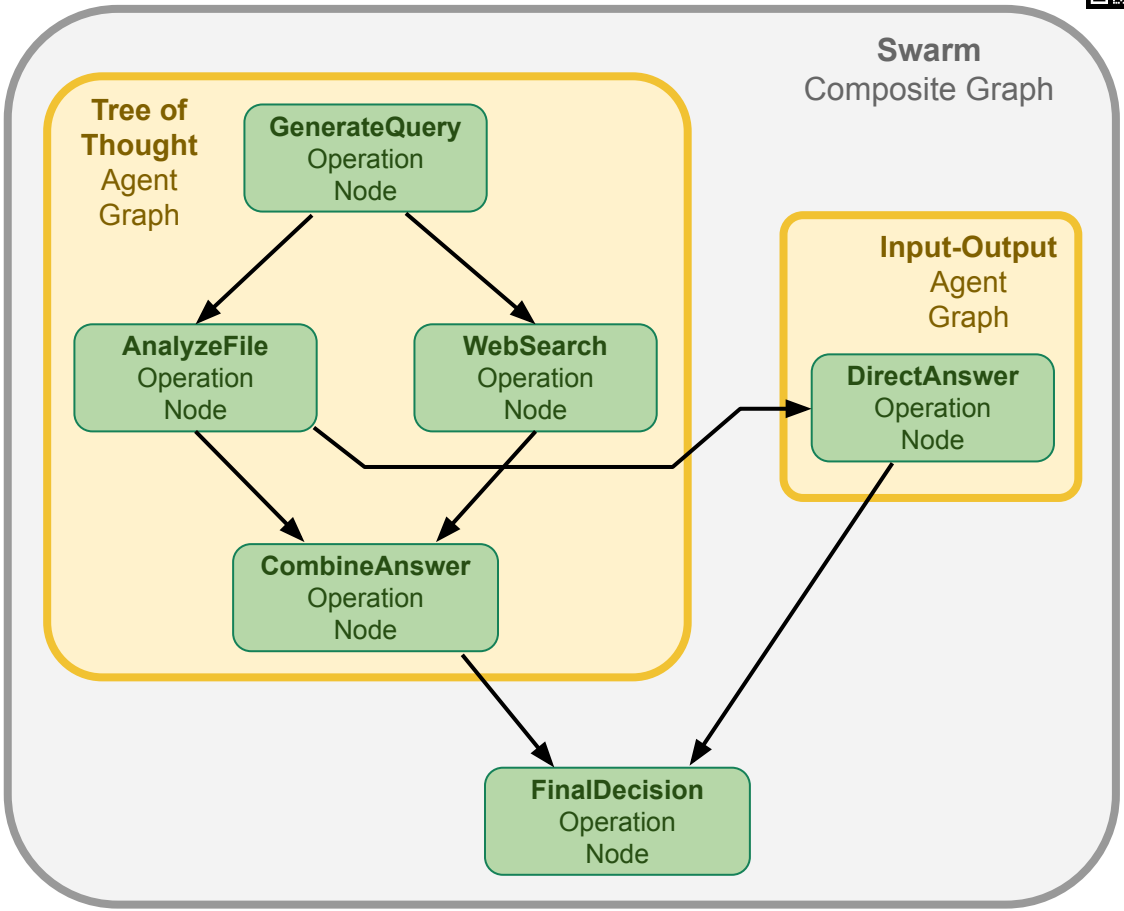
  Sample input $x \sim D_X$.

  $y \leftarrow G^P(x)$ following Algorithm 1.

  $h_n \leftarrow h_n \cup \{((z_n, x), f_n^{p_n}(z_n, x))\}$ for all $n \in N$.

  $p_n \leftarrow I(h_n, p_n, d_n)$, for all $n \in N$.

end while



3. Algorithms

Node Optimization

Improve individual operations

Algorithm 3 Node Optimization

Require: A parameterized graph $G^P = (N, E, F^P, o)$, natural language function descriptions $D = \{d_n\}_{n \in N}$, and a distribution of inputs D_X.

Initialize p_n for all $n \in N$.

Initialize $h_n \leftarrow \emptyset$ for all $n \in N$.

while terminate condition not met **do**

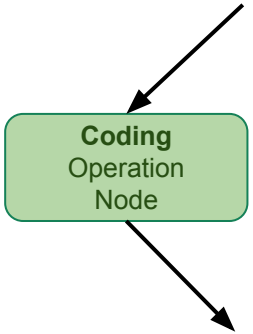
  Sample input $x \sim D_X$.

  $y \leftarrow G^P(x)$ following Algorithm 1.

  $h_n \leftarrow h_n \cup \{((z_n, x), f_n^{p_n}(z_n, x))\}$ for all $n \in N$.

  $p_n \leftarrow I(h_n, p_n, d_n)$, for all $n \in N$.

end while



3. Algorithms

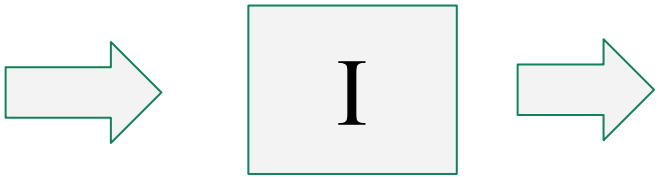
Node Optimization

Improve individual operations

Prompt:
Write a Python function.

Input:
…

Output:
…

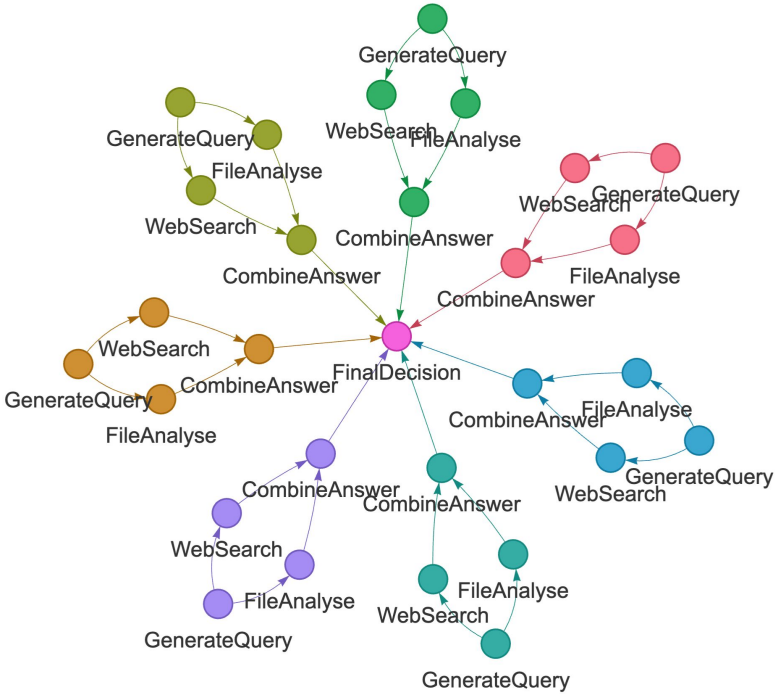


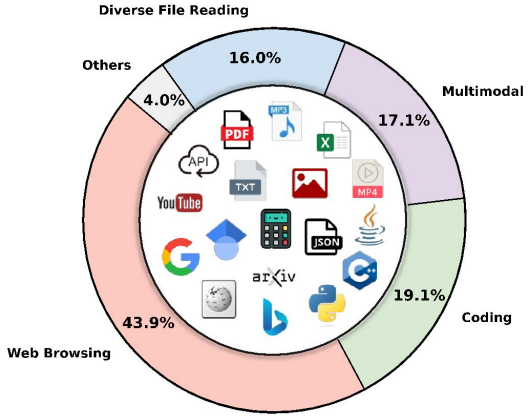
New prompt:
Write a Python function.
For example,

```python
def max_k(arr: list[int], k: int) -> list[int]:
    return sorted(arr, reverse=True)[:k]
```

4. Composing Agents with Graphs

Open-Ended Human Assistant (GAIA)





Tool calls as nodes in the graph

Method	Level 1	Level 2	Level 3	Avg.
GPT-3.5	7.55	4.65	0	4.85
GPT-4	15.09	2.33	0	6.06
GPT-4-Turbo	20.75	5.81	0	9.70
AutoGPT	13.21	0	3.85	4.85
GPTSwarm	**30.56**$\pm$3.25	**20.93**$\pm$1.27	**3.85**$\pm$2.43	**18.45**
Improvement	47.3%↑	260.2%↑	0.0%	90.2%↑
GPT4 with Plugins*	30.30	9.70	0	14.6

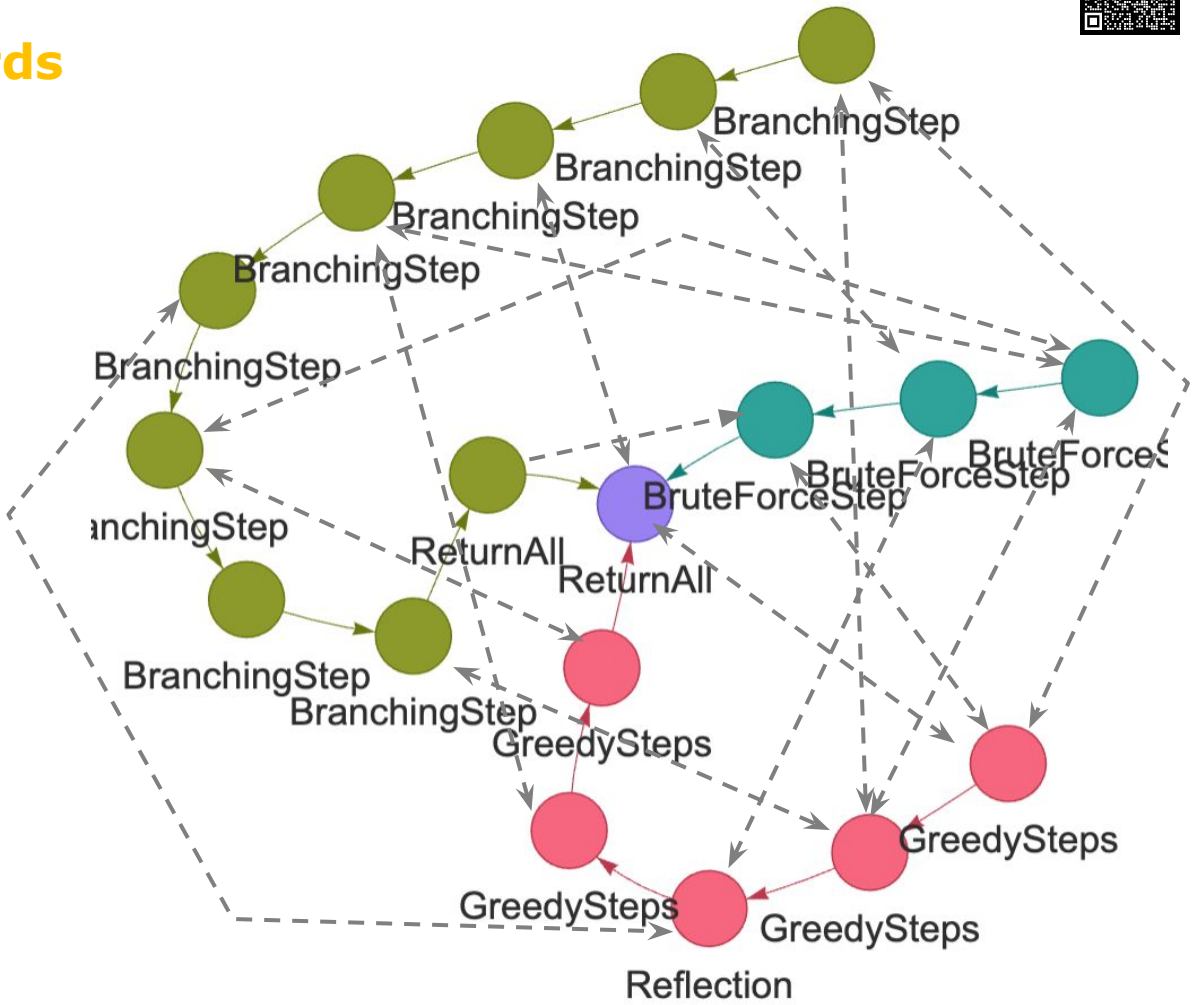
5. Edge Optimization

Solving Mini Crosswords

We are optimizing a swarm of three agents

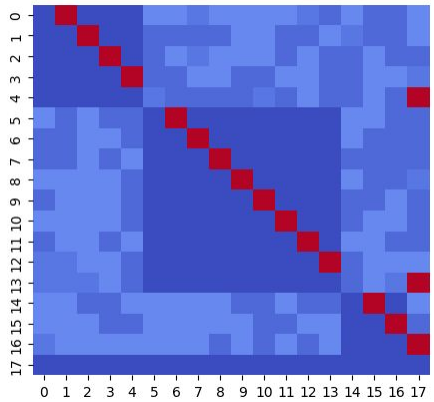
1) TOT
2) Reflexion
3) COT

D	A	D			S	E	N	D
O		E	A	S	T			A
W		A			I	T	S	Y
N	E	R	F		N		T	
	A		A	R	K		U	
	S		T		S	Y	N	C
M	E	S	H			A		A
A			E	V	E	R		R
N	E	A	R			D		D

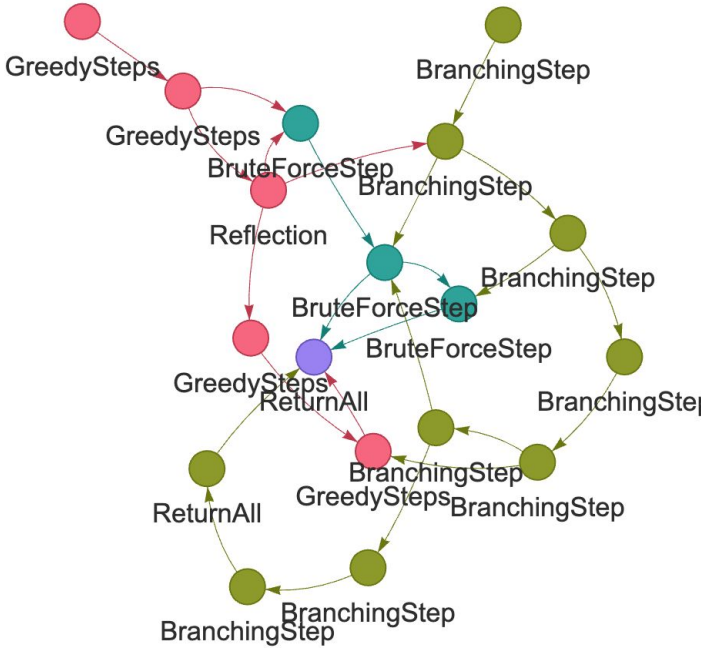


6. Edge Optimization

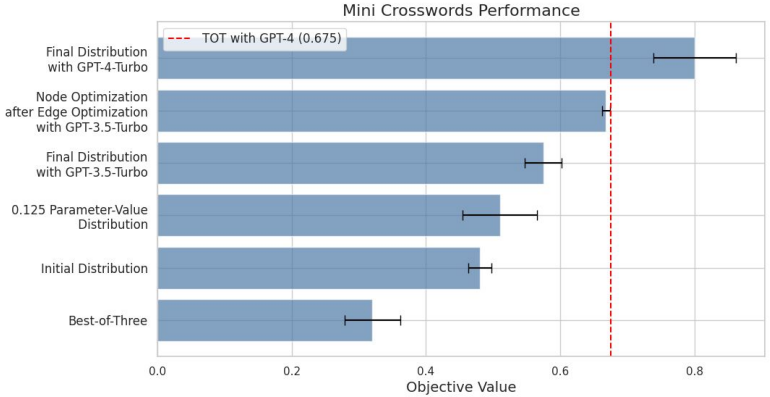
Solving Mini Crosswords



Adjacency matrix during optimization



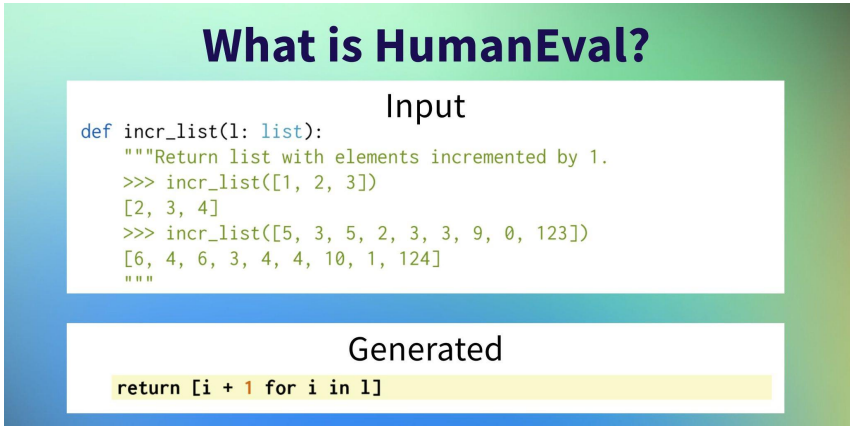
The optimized graph

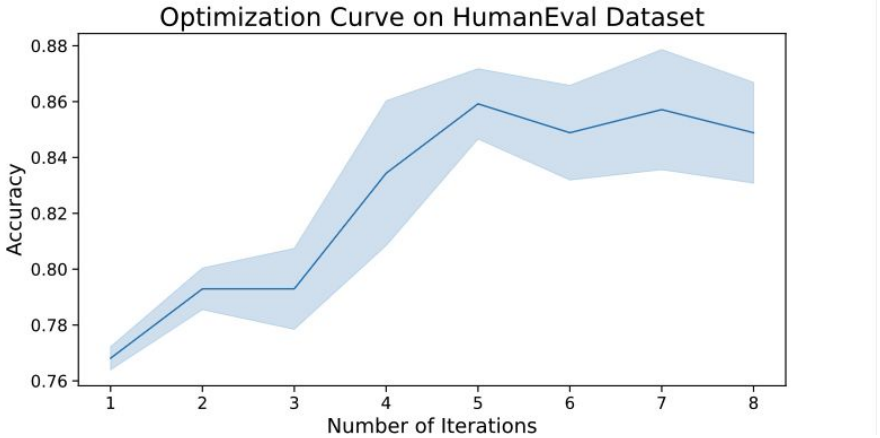


GPTSwarm Surpasses TOT

7. Node Optimization:

Code Generation on HumanEval





- ReAct-style agent
- We perform automatic prompt optimization at each node
- Selectively including positive input-output pairs on the node-level as few-shot examples
- We increased the pass rate @1 accuracy from 76% to 88%

What’s next?

- Better graph optimizers
- Joint node and edge optimization
- Scaling to larger graphs

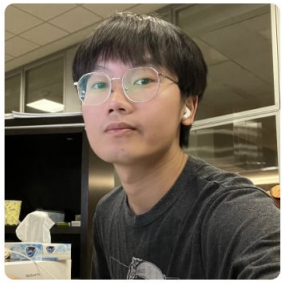
Thanks and Q&A

King Abdullah University of Science and Technology

Università della Svizzera italiana

USI/SUPSI Istituto Dalle Molle di studi sull'intelligenza artificiale IDSIA

Scuola universitaria professionale della Svizzera italiana SUPSI



Mingchen Zhuge

PhD Student, *KAUST AI Initiative*

Wenyi Wang

PhD Student, *KAUST AI Initiative*

Louis Kirsch

PhD Student, *IDSIA*

Francesco Faccio

PostDoc, *IDSIA*

Dmitrii Khizbullin

Reseach Engineer Lead, *KAUST AI Initiative*

Jürgen Schmidhuber

Director of *KAUST AI Initiative* and Scientific Director of *IDSIA*