🏔️ Position: Scaling LLM Agents Requires Asymptotic Analysis with LLM Primitives

Elliot Meyerson & Xin Qiu
Cognizant AI Lab

ICML 2025, arxiv.org/abs/2502.04358

LLM Agents

- **Agent:** Encapsulated program with specific role and driven by an LLM
- **Multi-agent System:** Program/system decomposed into multiple agents
- **Anthropomorphism:** Decomposition is often intuitive, mirroring human organizations
- ***How close are such organizations to optimal?***
- **NOT CLOSE AT ALL!**

🏔️ Asymptotic Analysis with LLM Primitives

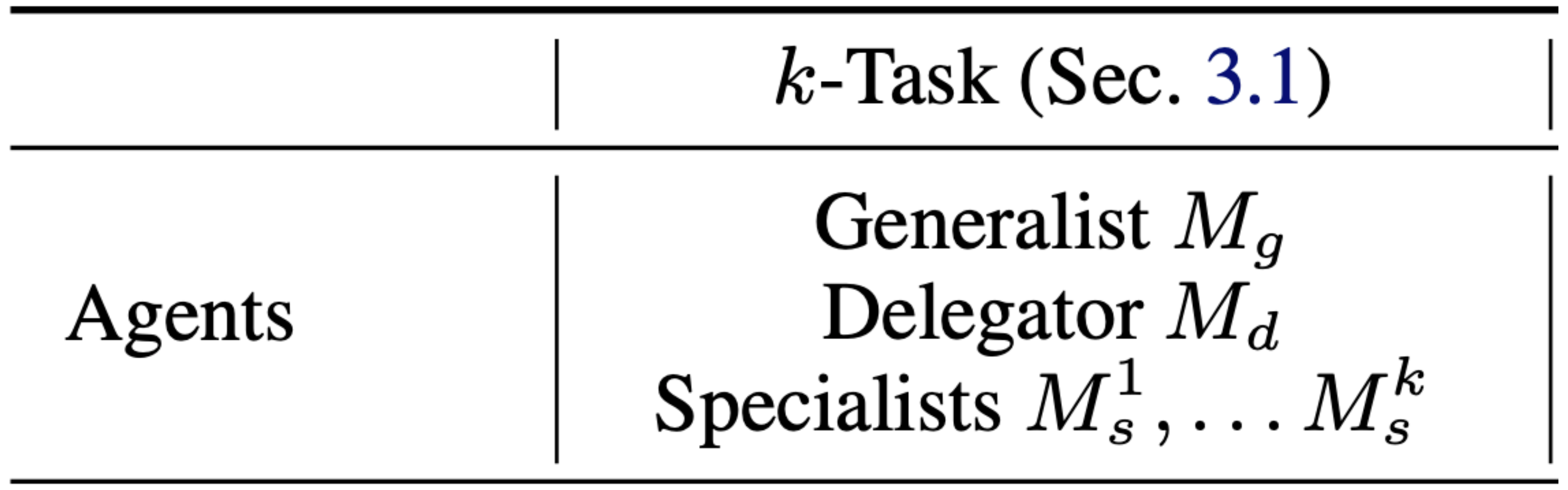
- **Asymptotic Analysis:** Understand fundamental scaling behavior
- **Language-based Algorithm (LbA):** A program where some steps are performed with LLMs.
- **Capabilities Set:** What a particular LLM can reliably do.
- **LLM Primitive:** The cost of a single forward pass of a particular LLM.

🏔️ AALPs -> The Right Level of Abstraction

Examples

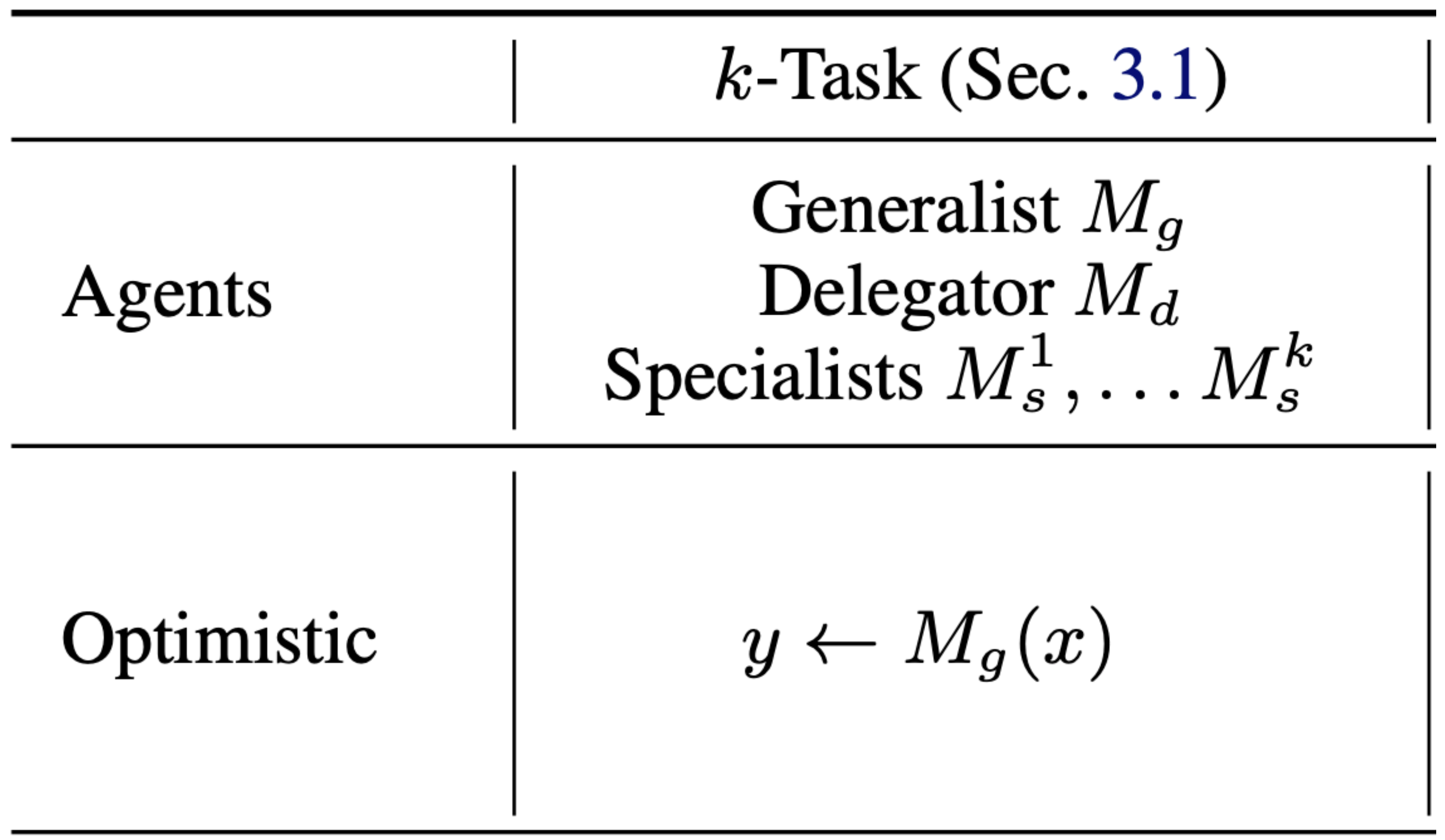
Agents	

Examples



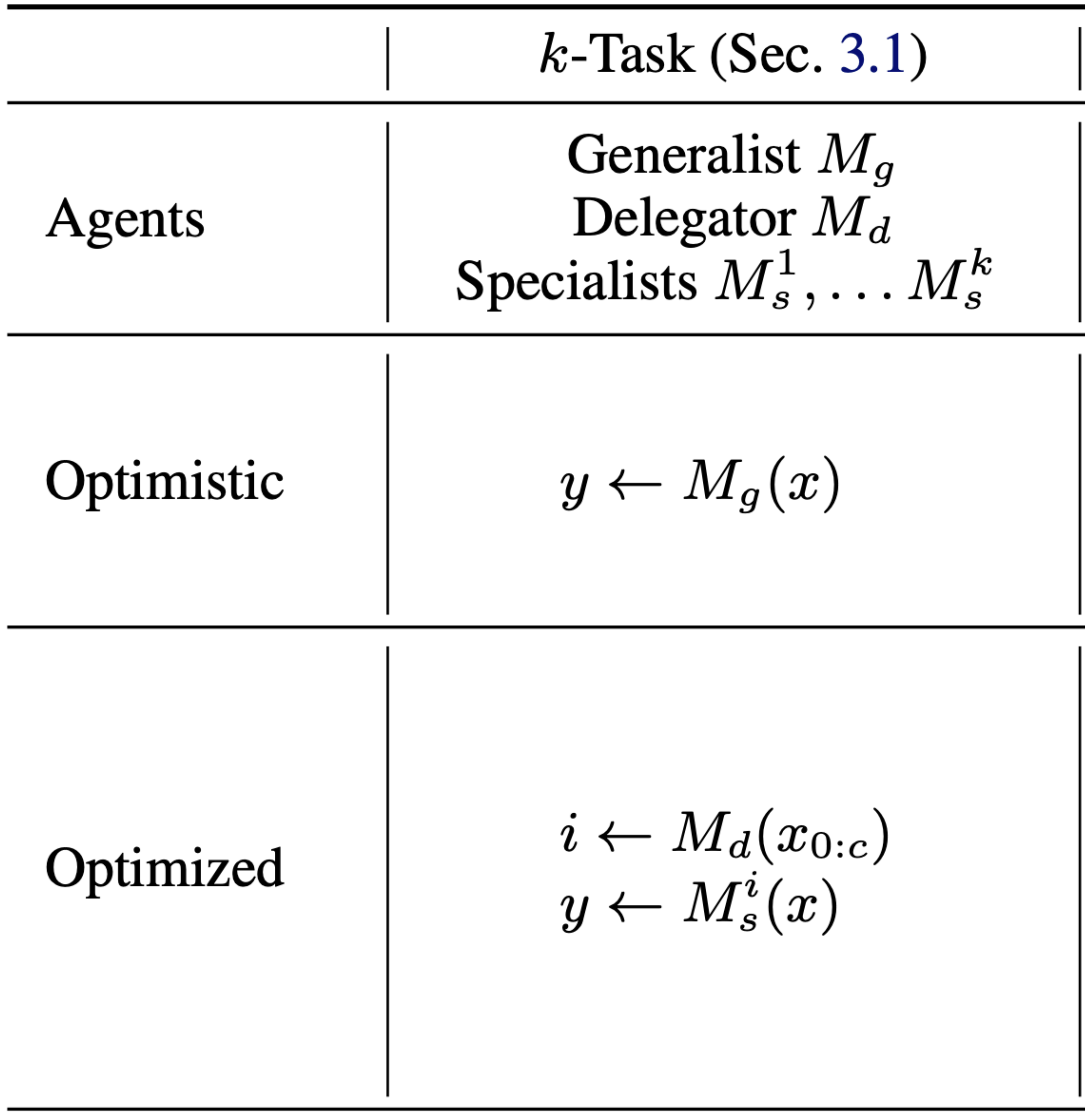
	k-Task (Sec. 3.1)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$

Examples



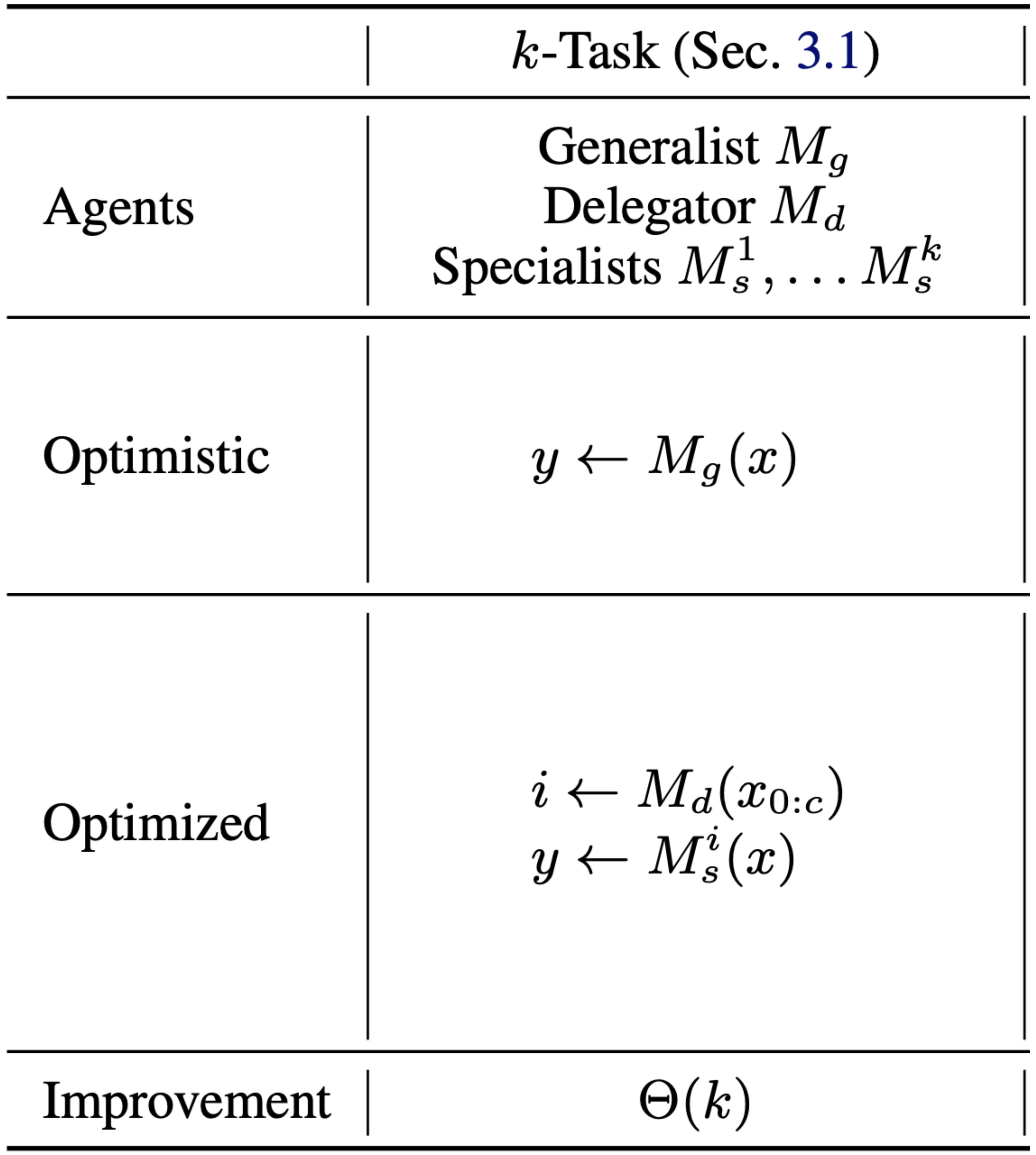
	k-Task (Sec. 3.1)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$
Optimistic	$y \leftarrow M_g(x)$

Examples



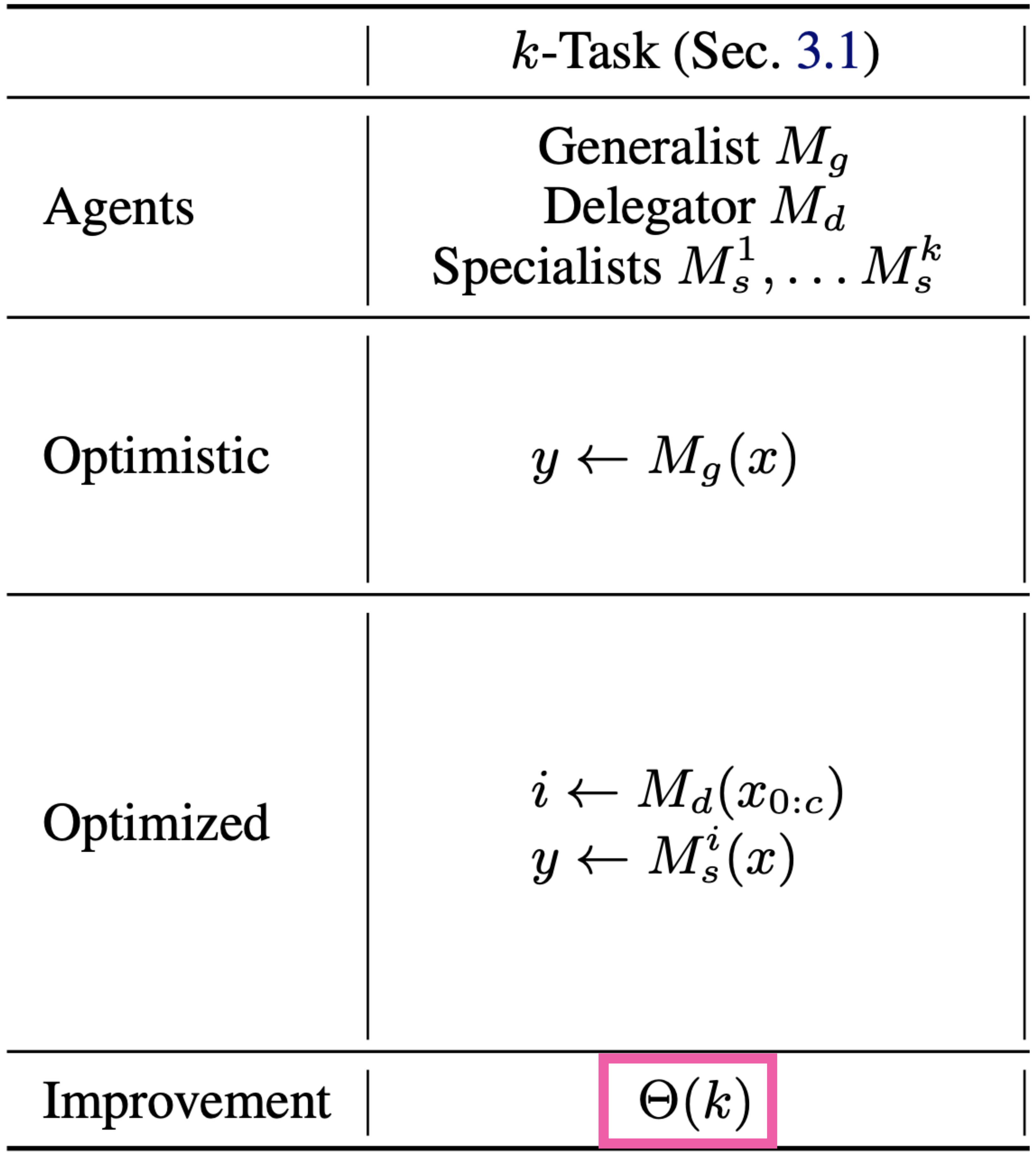
	k-Task (Sec. 3.1)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$
Optimistic	$y \leftarrow M_g(x)$
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$

Examples



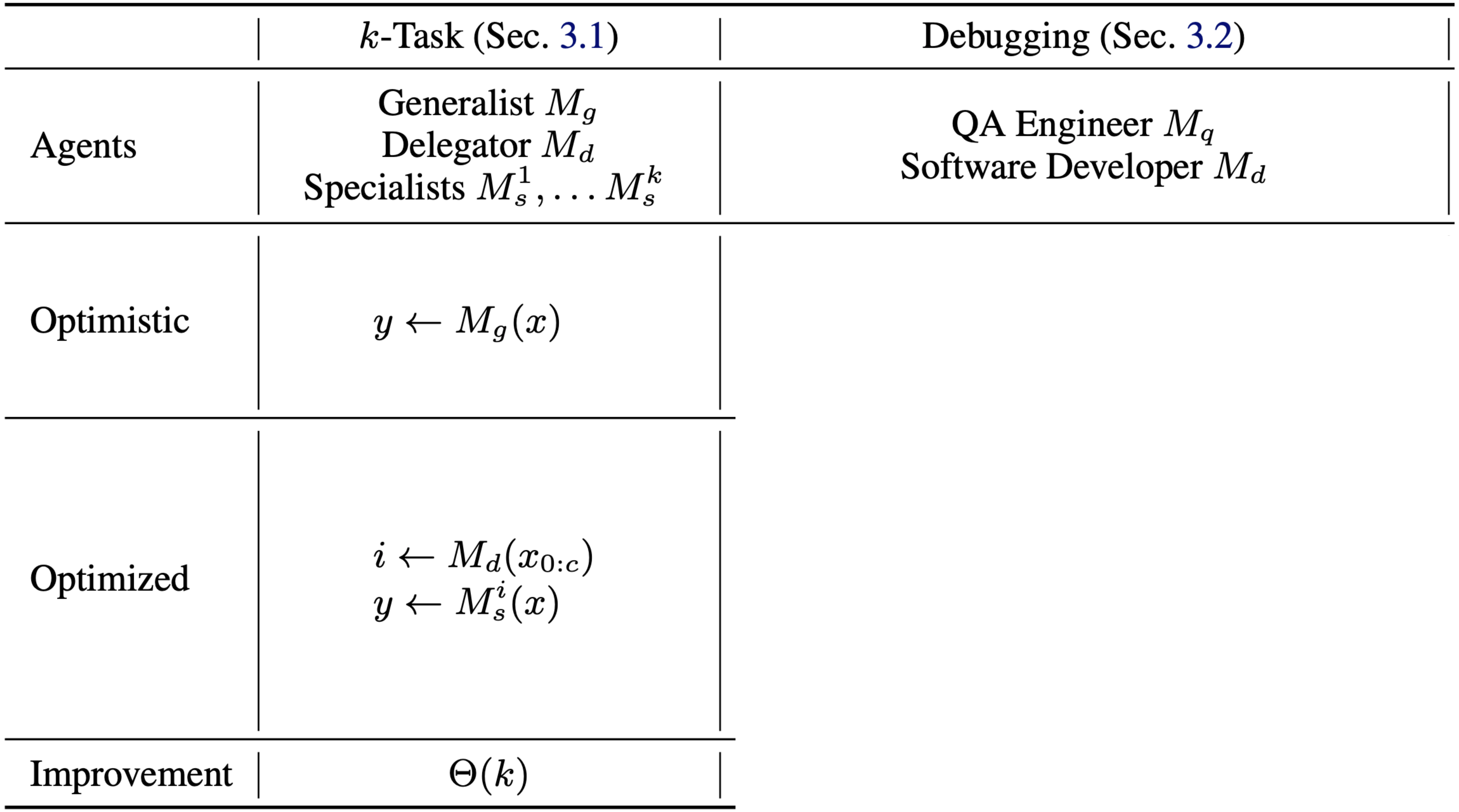
	k-Task (Sec. 3.1)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$
Optimistic	$y \leftarrow M_g(x)$
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$
Improvement	$\Theta(k)$

Examples



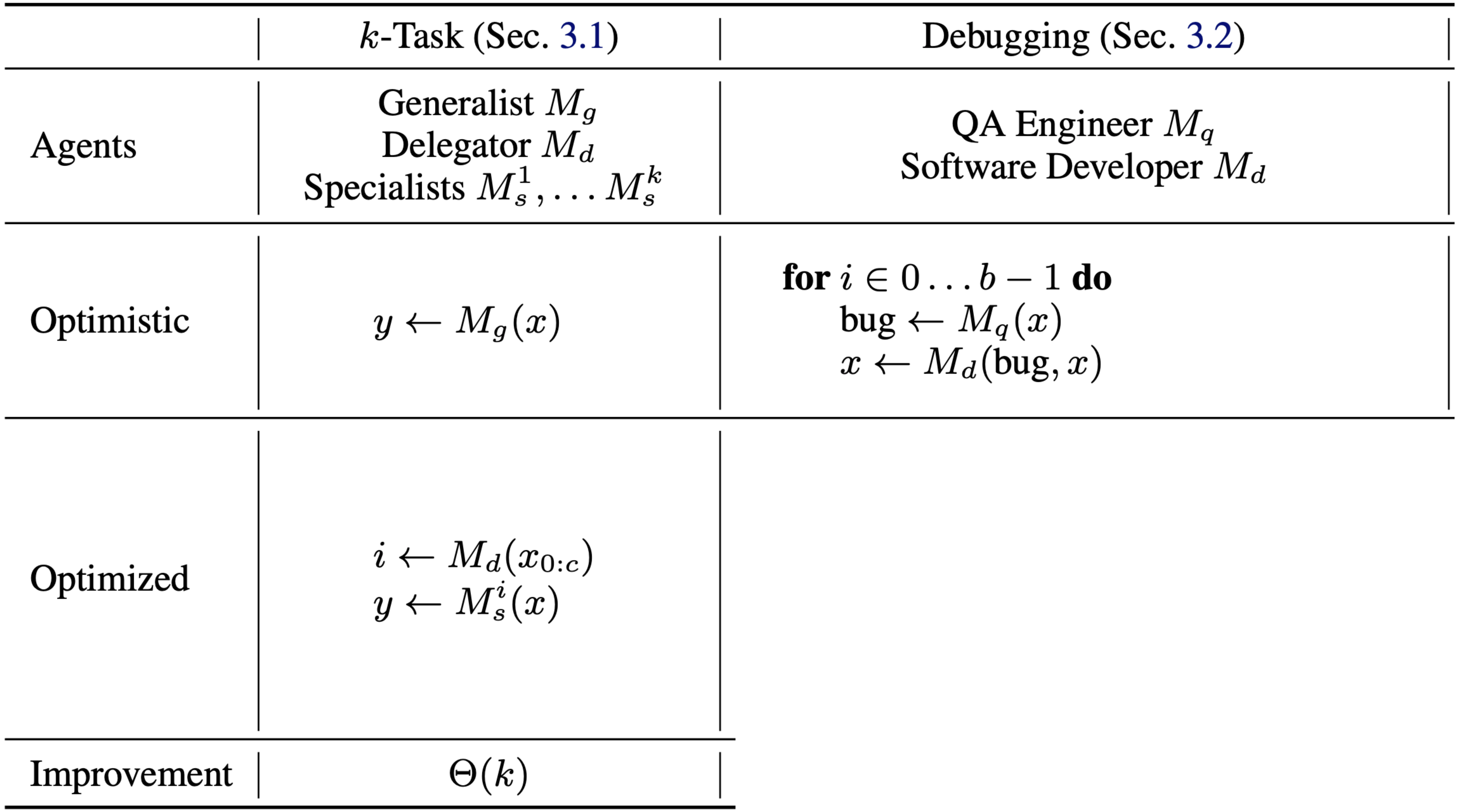
	k-Task (Sec. 3.1)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$
Optimistic	$y \leftarrow M_g(x)$
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$
Improvement	$\boxed{\Theta(k)}$

Examples



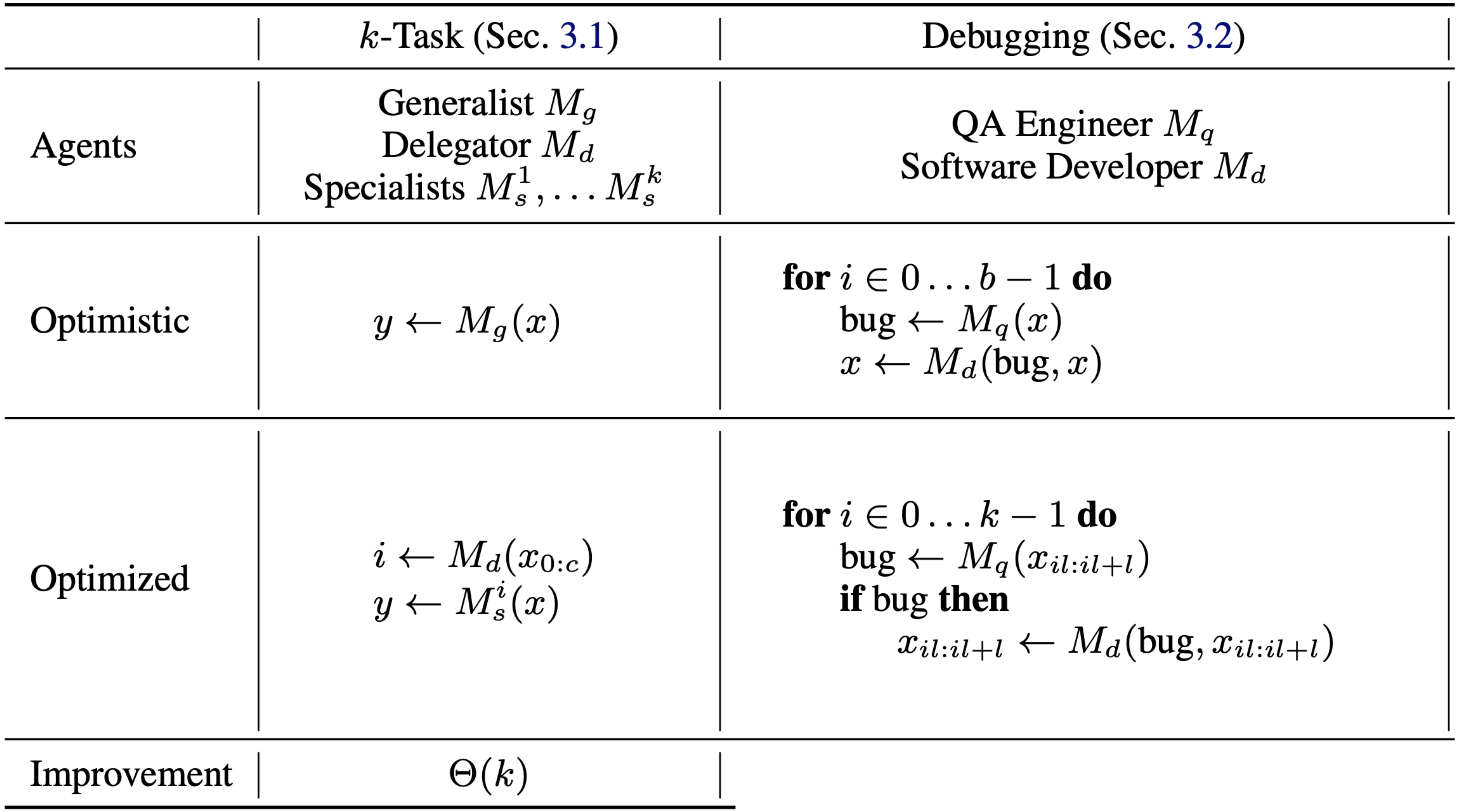
	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$	QA Engineer M_q Software Developer M_d
Optimistic	$y \leftarrow M_g(x)$	
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$	
Improvement	$\Theta(k)$	

Examples



	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d
Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do** bug $\leftarrow M_q(x)$ $x \leftarrow M_d(\text{bug}, x)$
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$	
Improvement	$\Theta(k)$	

Examples

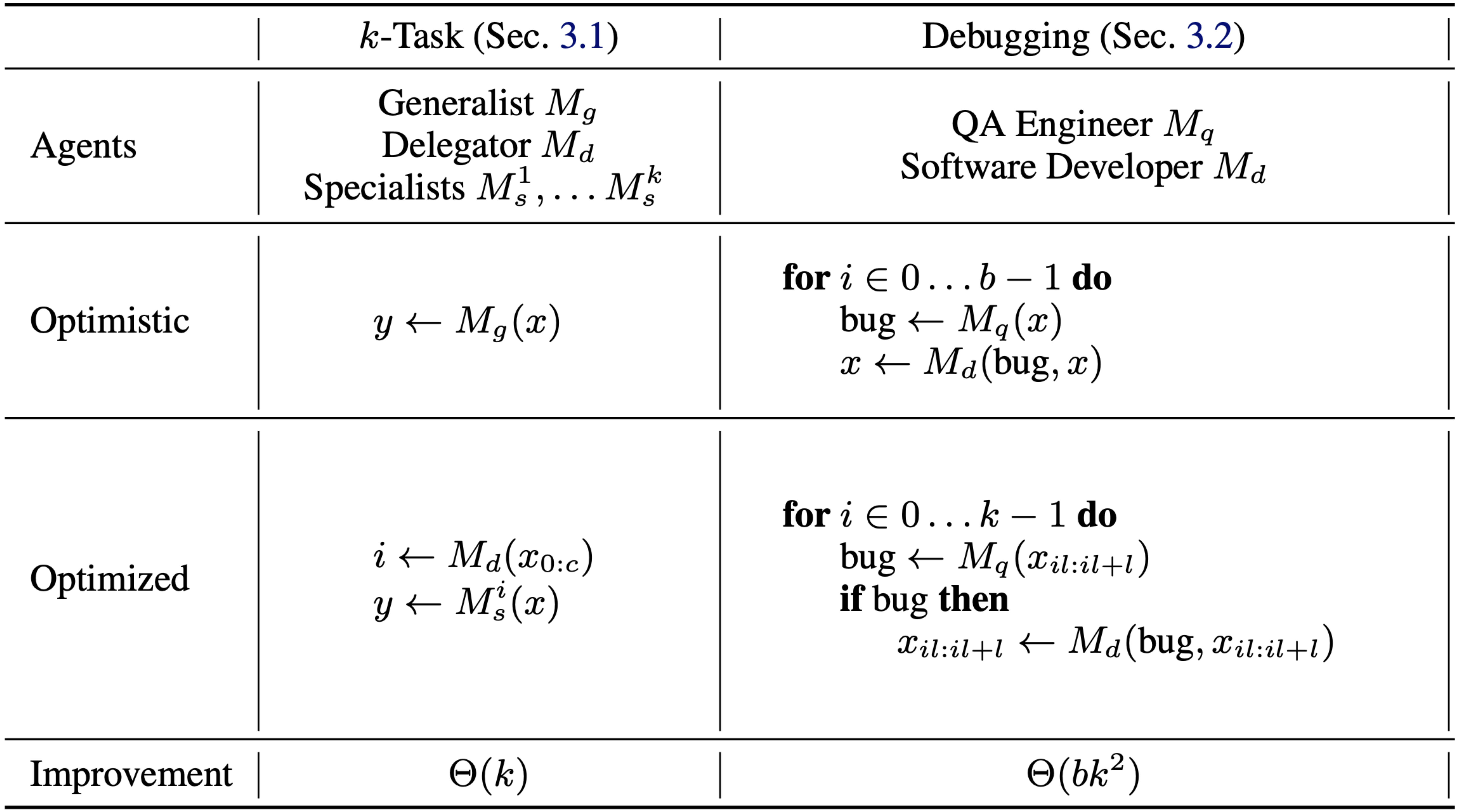


	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d
Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do** bug $\leftarrow M_q(x)$ $x \leftarrow M_d(\text{bug}, x)$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \ldots k-1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$

Improvement	$\Theta(k)$	

Examples

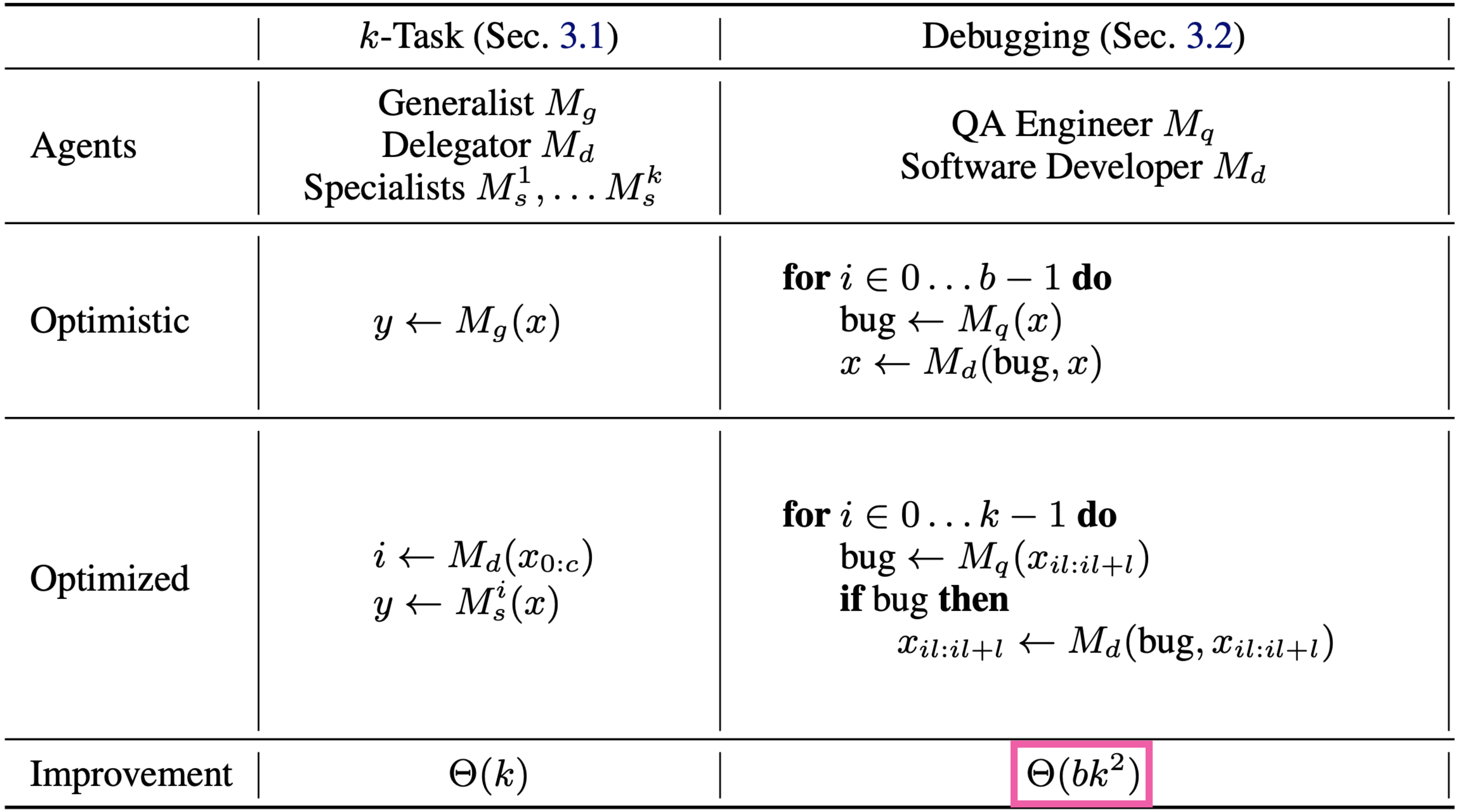


	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$	QA Engineer M_q Software Developer M_d
Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \dots b-1$ **do** bug $\leftarrow M_q(x)$ $x \leftarrow M_d(\text{bug}, x)$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \dots k-1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$

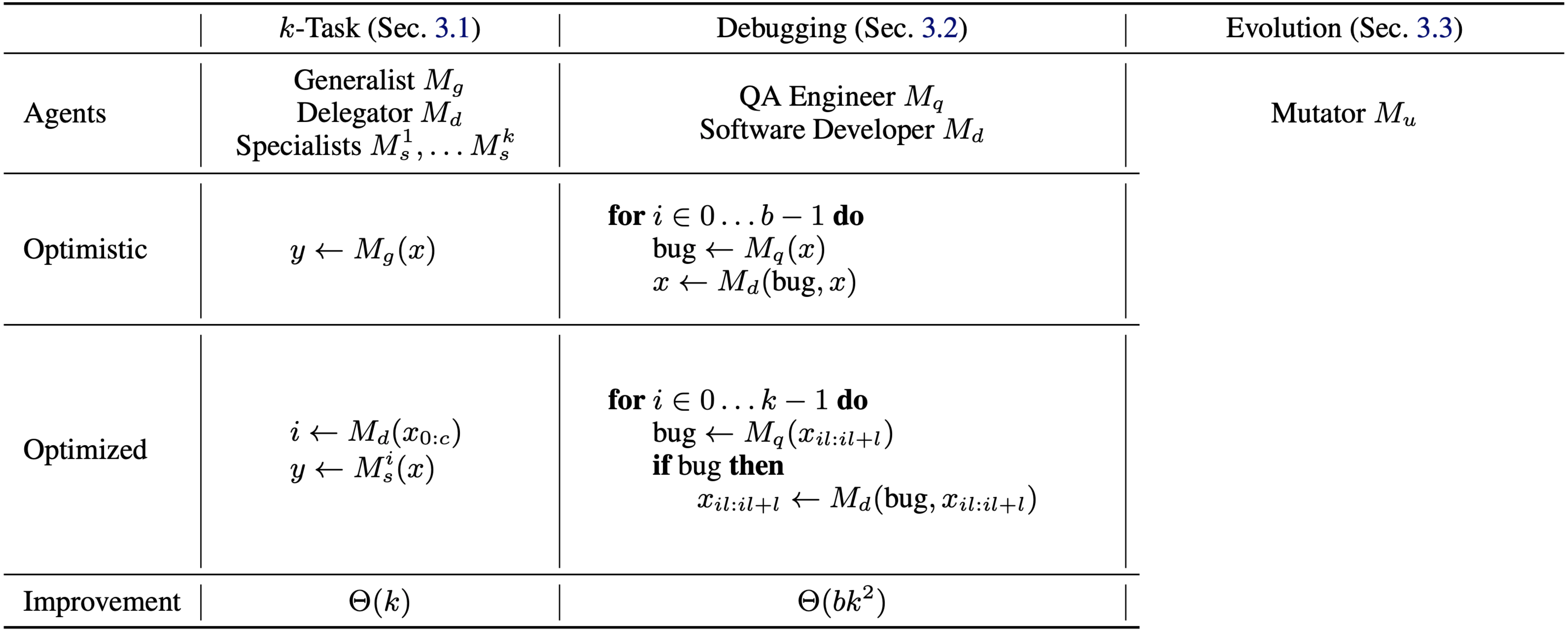
Improvement	$\Theta(k)$	$\Theta(bk^2)$

Examples



	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$	QA Engineer M_q Software Developer M_d
Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \dots b-1$ **do** $\quad$ bug $\leftarrow M_q(x)$ $\quad x \leftarrow M_d(\text{bug}, x)$
Optimized	$i \leftarrow M_d(x_{0:c})$ $y \leftarrow M_s^i(x)$	**for** $i \in 0 \dots k-1$ **do** $\quad$ bug $\leftarrow M_q(x_{il:il+l})$ $\quad$ **if** bug **then** $\qquad x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$
Improvement	$\Theta(k)$	$\boxed{\Theta(bk^2)}$

Examples



	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do**
		    bug $\leftarrow M_q(x)$
		    $x \leftarrow M_d(\text{bug}, x)$	

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \ldots k-1$ **do**
		    bug $\leftarrow M_q(x_{il:il+l})$
		    if bug **then**
		        $x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$	

Improvement	$\Theta(k)$	$\Theta(bk^2)$	

Examples

	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do**
		bug $\leftarrow M_q(x)$
		$x \leftarrow M_d(\text{bug}, x)$
			while $f(x) < k$ **do**
			$x' \leftarrow M_u(x)$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \ldots k-1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$	

Improvement	$\Theta(k)$	$\Theta(bk^2)$	

Examples

	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \dots b - 1$ **do**
		bug $\leftarrow M_q(x)$
		$x \leftarrow M_d(\text{bug}, x)$
			while $f(x) < k$ **do**
			$x' \leftarrow M_u(x)$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \dots k - 1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$
			$i \leftarrow 0$
			while $f(x) < k$ **do**
			$x' \leftarrow x$
			$x'_{il:il+l} \leftarrow M_u(x_{il:il+l})$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$
			$i \leftarrow i + 1$

Improvement	$\Theta(k)$	$\Theta(bk^2)$	

Examples

	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do**
		  bug $\leftarrow M_q(x)$
		  $x \leftarrow M_d(\text{bug}, x)$
			while $f(x) < k$ **do**
			  $x' \leftarrow M_u(x)$
			  if $f(x') > f(x)$ **then**
			    $x \leftarrow x'$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \ldots k-1$ **do**
		  bug $\leftarrow M_q(x_{il:il+l})$
		  if bug **then**
		    $x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$
			$i \leftarrow 0$
			while $f(x) < k$ **do**
			  $x' \leftarrow x$
			  $x'_{il:il+l} \leftarrow M_u(x_{il:il+l})$
			  if $f(x') > f(x)$ **then**
			    $x \leftarrow x'$
			    $i \leftarrow i+1$

Improvement	$\Theta(k)$	$\Theta(bk^2)$	$\Theta(2^k k)$

Examples

	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \dots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \dots b - 1$ **do**
		bug $\leftarrow M_q(x)$
		$x \leftarrow M_d(\text{bug}, x)$
			while $f(x) < k$ **do**
			$x' \leftarrow M_u(x)$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \dots k - 1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$
			$i \leftarrow 0$
			while $f(x) < k$ **do**
			$x' \leftarrow x$
			$x'_{il:il+l} \leftarrow M_u(x_{il:il+l})$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$
			$i \leftarrow i + 1$

Improvement	$\Theta(k)$	$\Theta(bk^2)$	$\boxed{\Theta(2^k k)}$

Examples

	k-Task (Sec. 3.1)	Debugging (Sec. 3.2)	Evolution (Sec. 3.3)
Agents	Generalist M_g Delegator M_d Specialists $M_s^1, \ldots M_s^k$	QA Engineer M_q Software Developer M_d	Mutator M_u

Optimistic	$y \leftarrow M_g(x)$	**for** $i \in 0 \ldots b-1$ **do**
		bug $\leftarrow M_q(x)$
		$x \leftarrow M_d(\text{bug}, x)$
			while $f(x) < k$ **do**
			$x' \leftarrow M_u(x)$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$

Optimized	$i \leftarrow M_d(x_{0:c})$
	$y \leftarrow M_s^i(x)$
		for $i \in 0 \ldots k-1$ **do**
		bug $\leftarrow M_q(x_{il:il+l})$
		if bug **then**
		$x_{il:il+l} \leftarrow M_d(\text{bug}, x_{il:il+l})$
			$i \leftarrow 0$
			while $f(x) < k$ **do**
			$x' \leftarrow x$
			$x'_{il:il+l} \leftarrow M_u(x_{il:il+l})$
			if $f(x') > f(x)$ **then**
			$x \leftarrow x'$
			$i \leftarrow i+1$

Improvement	$\Theta(k)$	$\Theta(bk^2)$	$\boxed{\Theta(2^k k)}$

🏔️ More exercises in the paper!

Research Directions

- Identifying useful assumptions
- Stochasticity and Error-correction
- Asynchrony and Distributed Algorithms
- Automatic Decomposition
- Extensions to Other Modalities
- Online Learning and Adaptation
- Ethics and Sentience
- ...

Research Directions

- **Identifying useful assumptions**
- Stochasticity and Error-correction
- Asynchrony and Distributed Algorithms
- Automatic Decomposition
- Extensions to Other Modalities
- Online Learning and Adaptation
- Ethics and Sentience
- ...

Conclusion

- 🏔️ **AALPs:** A framework to understand and scale LLM-agent systems
- This paper is **an appeal to builders**
- This paper is **an invitation to algorithmists**
- **Don’t anthropomorphize your agents 🏔️ Scale them with AALPs!**
- Check out the full paper: arxiv.org/abs/2502.04358