Interactive Speculative Planning

Enhance Agent Efficiency through Co-design of System and User Interface

Wenyue Hua, Mengting Wan, Shashank Vadrevu, Ryan Nadel, Yongfeng Zhang, Chi Wang

Rutgers University, New Brunswick; Microsoft Research
University of California, Santa Barbara

Reddit · r/LangChain
40+ comments · 1 year ago

ed help in increasing the speed of my llm based ...

using OpenAI's embeddings as well, the ada-002. However, the responses are very **slow**. For little
plex questions it takes 20-30 seconds to respond. The size of the vectorstore is 62MB's only but ...

roaches to increase **speed** of LangChain **agent**? Takes	Oct 5, 2023
I **Agents**: too early, too expensive, too unreliable - Reddit	May 22, 2024
oving answers with editor **LLM agent** : r/LocalLLaMA	Jul 2, 2024
lype Behind **Agents**? : r/MachineLearning - Reddit	Jun 10, 2024

e results from www.reddit.com

scussions and forums

roaches to increase speed of LangChain agent? Takes

Reddit · r/LangChain · 10+ comments · 1 year ago

Summarization is Costing Me Thousands. 192 upvotes · 117 comments. r/LangChain icon.
ngChain. • 2 yr. ago · Need help in increasing the speed of my l... More ›

y slow Response from LLM based Q/A query engine

Stack Overflow · 2 answers · 1 year ago

emely slow agent with strange behaviour: After the tool finishes ...

GitHub · 9 months ago

See more →

r/MachineLearning · 9 mo. ago
madredditscientist

[D] AI Agents: too early, too expensive, too unreliable

Discussion

Reference: Full blog post

There has been a lot of hype about the promise of autonomous agent-based LLM workflows. By now, all major LLMs are capable of interacting with external tools and functions, letting the LLM perform sequences of tasks automatically.

But reality is proving more challenging than anticipated.

The WebArena leaderboard, which benchmarks LLMs agents against real-world tasks, shows that even the best-performing models have a success rate of only 35.8%.

Challenges in Practice

After seeing many attempts to AI agents, I believe it's too early, too expensive, too slow, too unreliable.
It feels like many AI agent startups are waiting for a model breakthrough that will start the race to productize
age

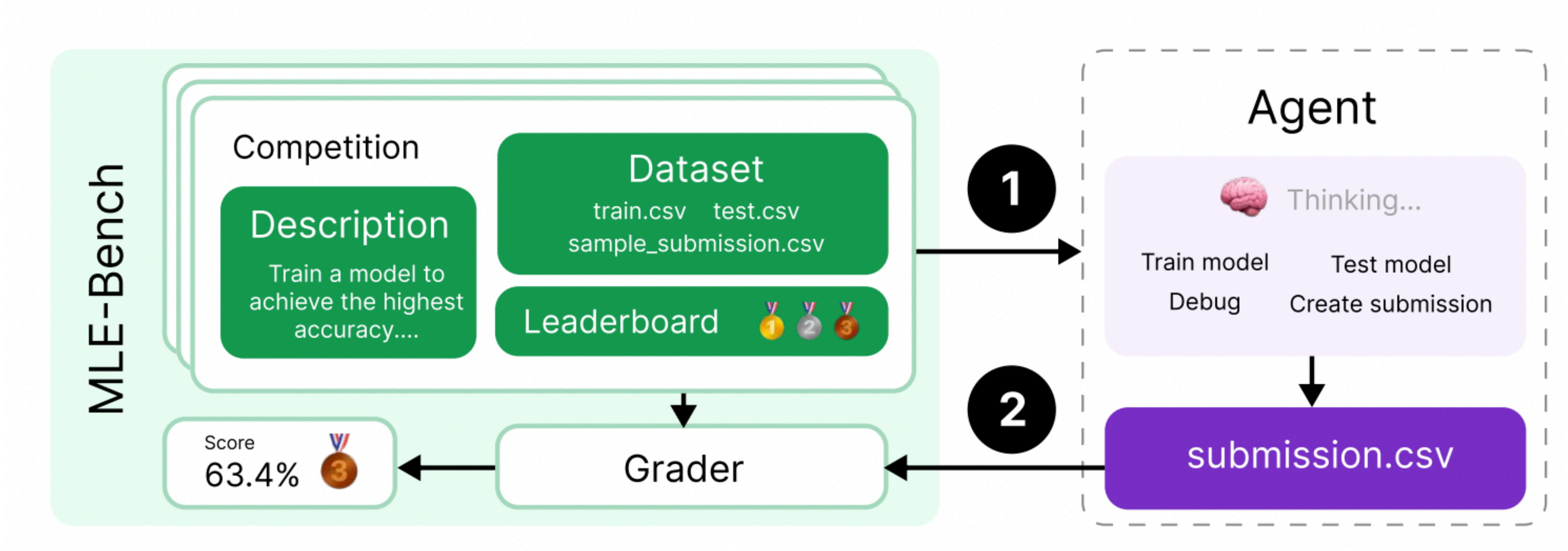
The Trilemma of Efficiency, Speed, and Performance in LLM Agents

Muthumari S
Global Head of AI Studio @ Brillio | Generative AI, Business Analytics, TRiSM

October 8, 2024

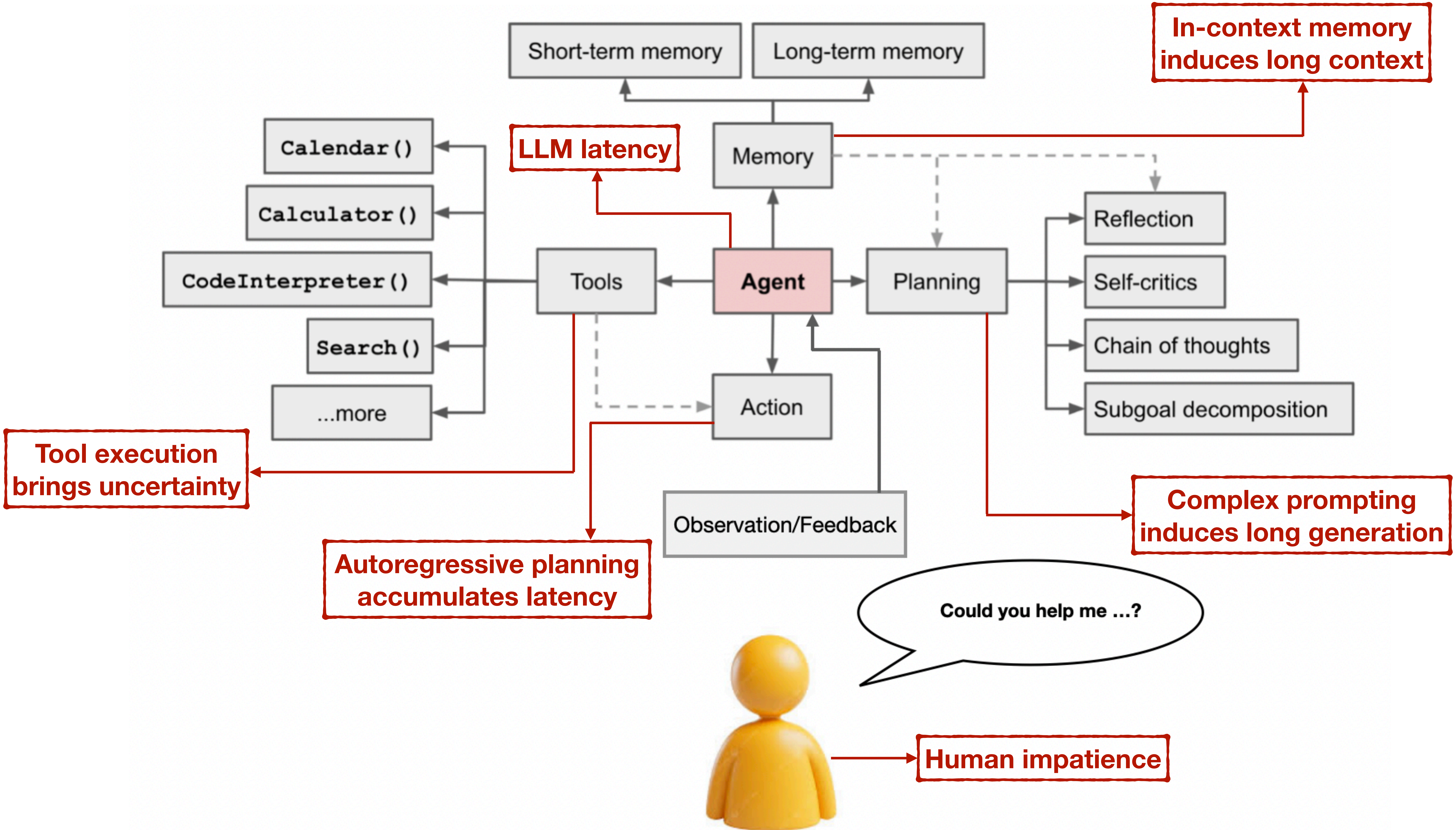
The emergence of **Large Language Models (LLMs)**, has revolutionized artific
intelligence across industries. These models, powered by billions of paramete
are the engines behind groundbreaking applications like **Generative AI, Natu
Language Processing (NLP)**, and intelligent automation. However, as businesses race to adopt **LLM agents** to automate complex tasks, a key challenge emerges—balancing the **trilemma** of **efficiency, speed, and performance**. In the context of LLM agents, achieving all three at once is akin
balancing a three-legged stool. Each factor plays a critical role in scaling AI-driven solutions, but optimizing for one can often lead to trade-offs with the

❖ In a text game simulating travel planning, one single request took me about 5 minutes to finish when doing experiments

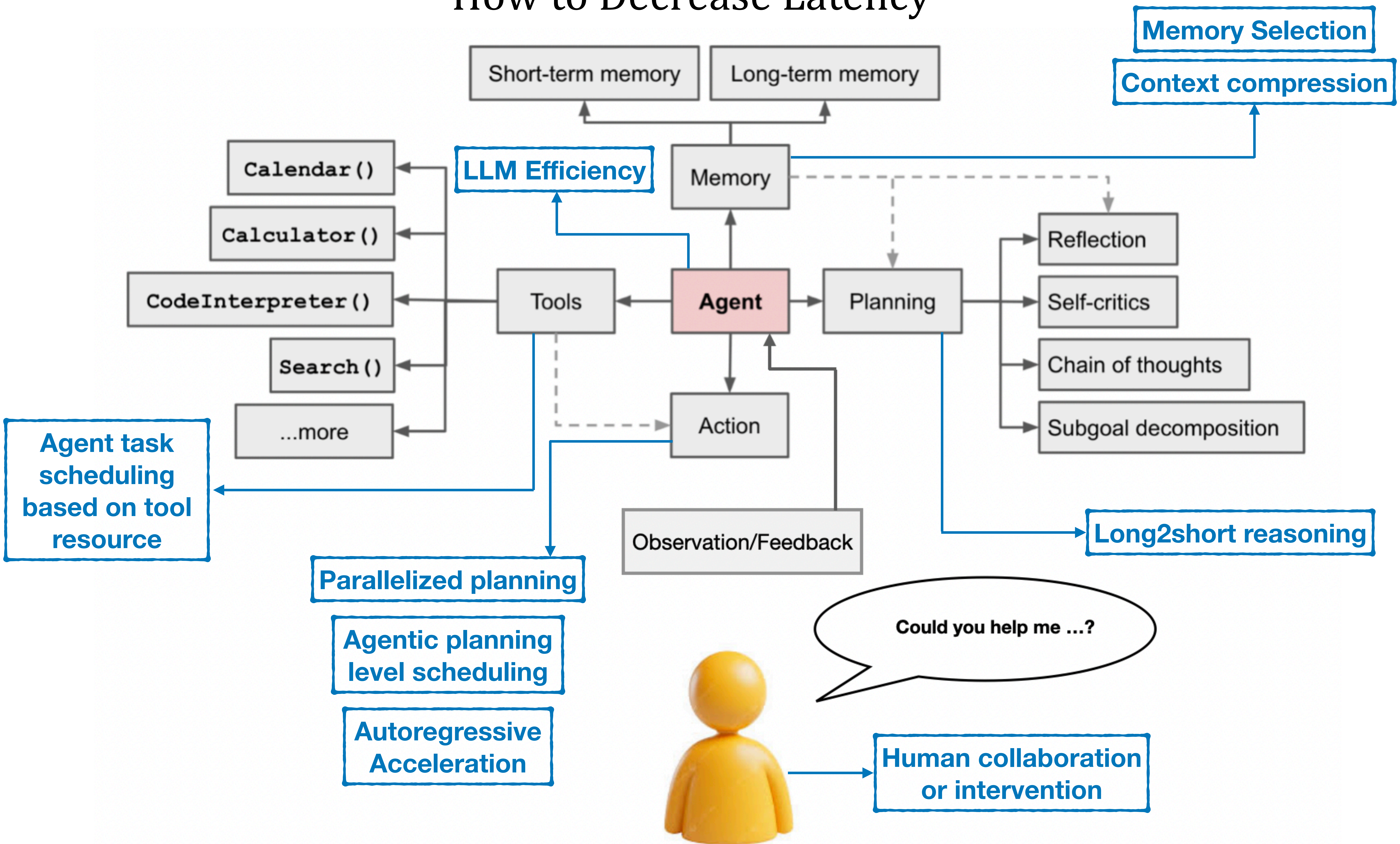


❖ MLE-bench evaluates an agent's ability to solve Kaggle challenges involving the design, building, and training of machine learning models on GPUs. O3-mini based agent is given 24 hours to develop a solution, scaled up to 100 hours in some experiments.

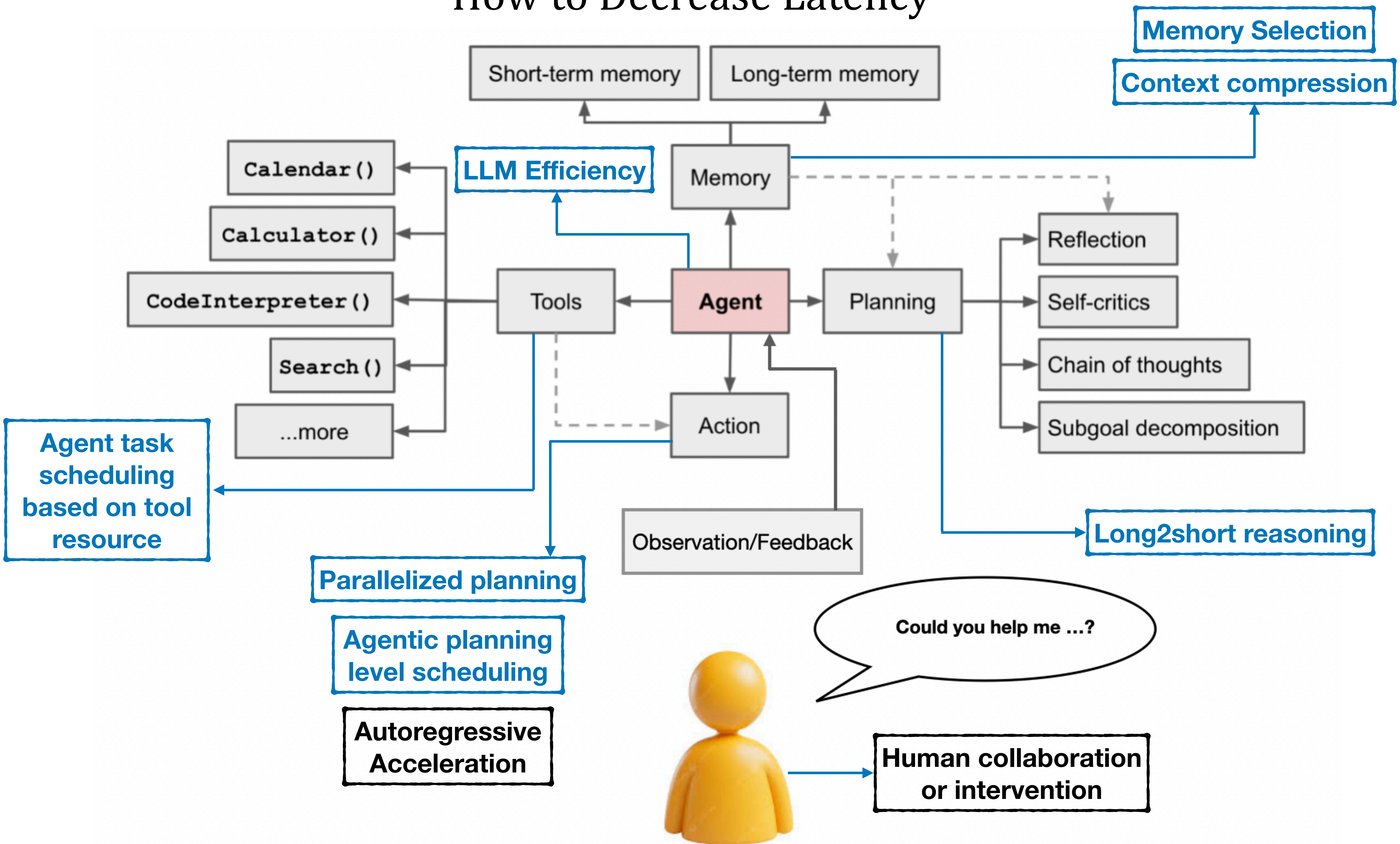
Sources of Latency



How to Decrease Latency

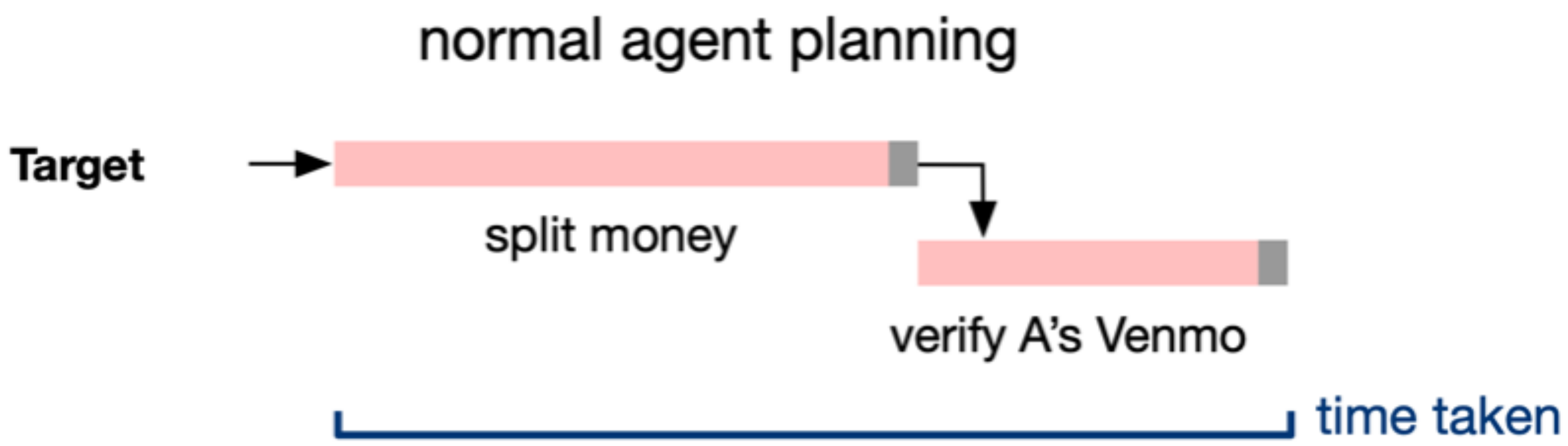


How to Decrease Latency



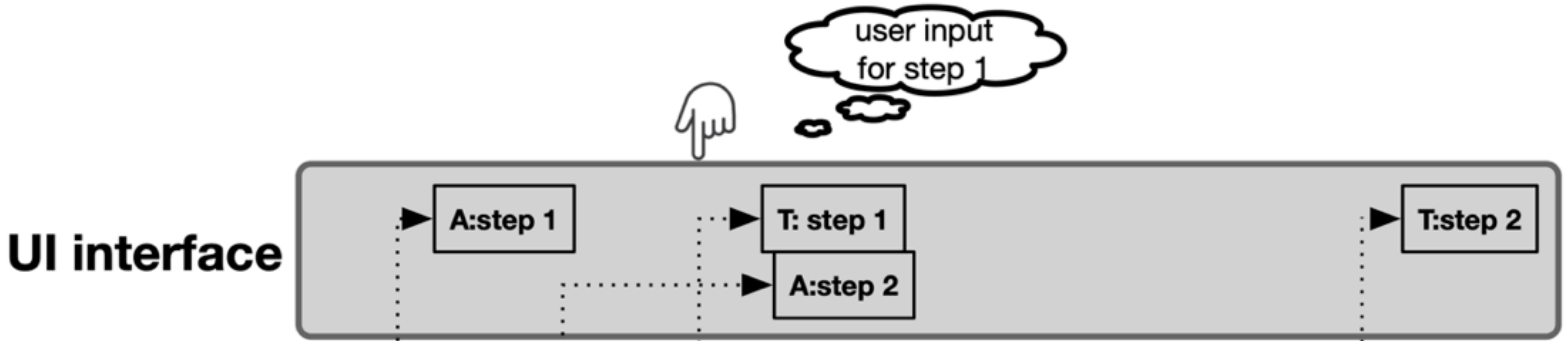
RQ: How to accelerate agent planning?

➢ How to accelerate agent planning by system design?



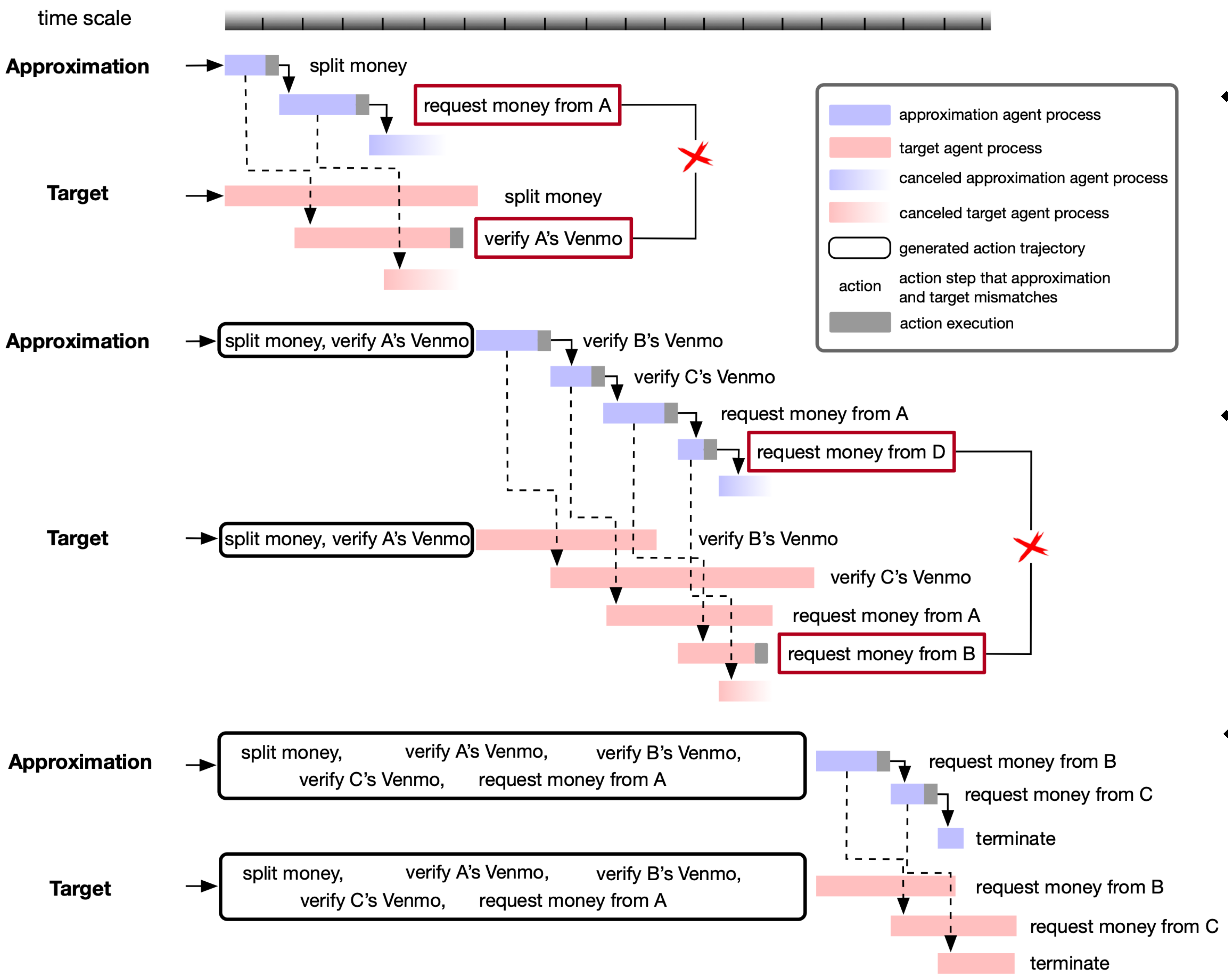
Use speculative planning to accelerate the autoregressive planning process

➢ How to enable active user interaction and leverage users to further accelerate agent planning?



Allow active user interaction: Interactive Speculative Planning

Main Algorithm for Efficiency: Speculative Planning

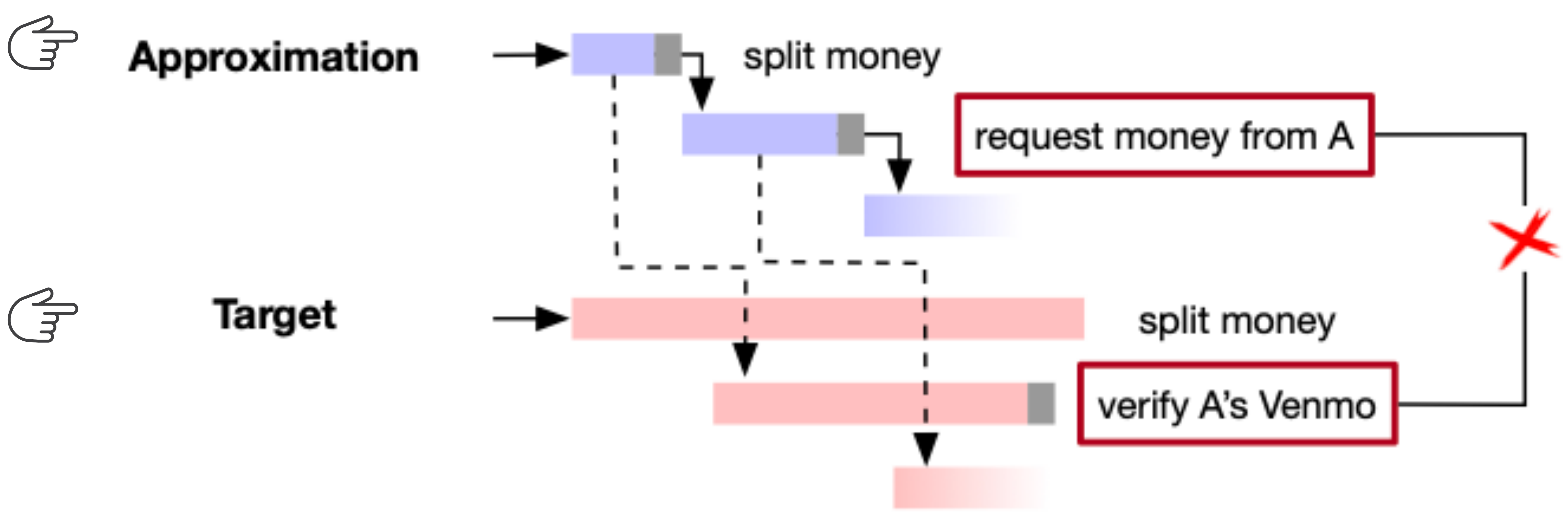


- ❖ Expedite agent planning by employing a fast and efficient approximation agent to resolve the task sequentially, with each step representing an action to be executed
- ❖ Target agent utilizes the result generated by the fast approximation agent as a prefix to generate the next step, rather than waiting for prefix steps from the slower target agent to be completed.
- ❖ Approximation agent is running sequentially while target is running in parallel.

Speculative Planning Algorithm Broken-down

an efficient but less capable approximation agent

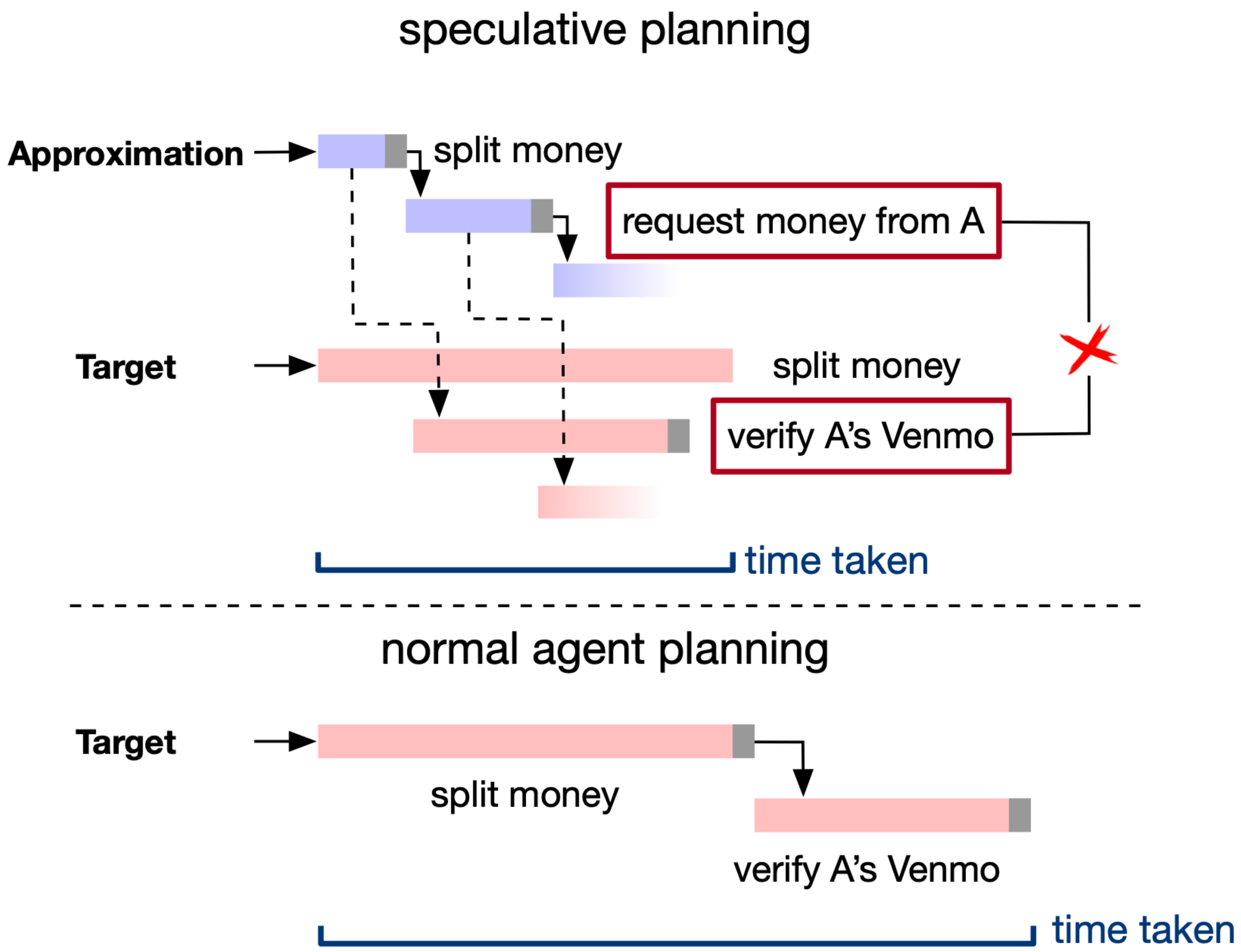
a slower but more powerful target agent



The approximation agent operates in a step-by-step manner, as shown by the sequential blue blocks, where each block represents the time taken to generate a step.

The target agent operates asynchronously: for each part of the plan generated by the approximation agent, the target agent calculates the next step in order to confirm whether steps generated by the approximation agent are correct, which processes are all in parallel as shown by the horizontally-overlapping pink blocks. Then, for each action in the plans generated by both agents, we check if they match to determine if the approximation agent is working correctly.

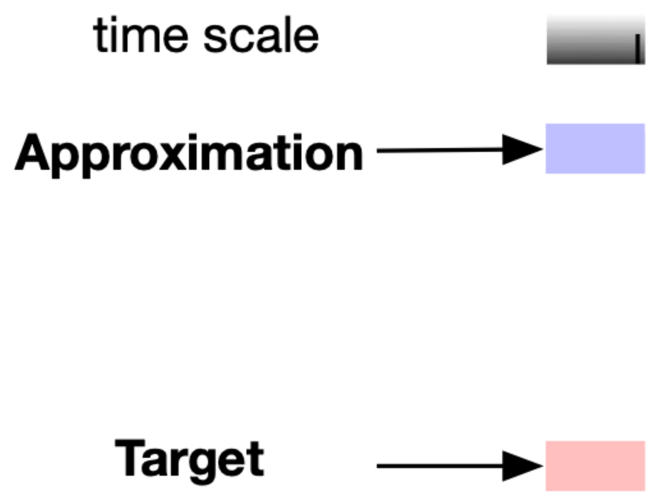
How time is saved in Speculative Planning?



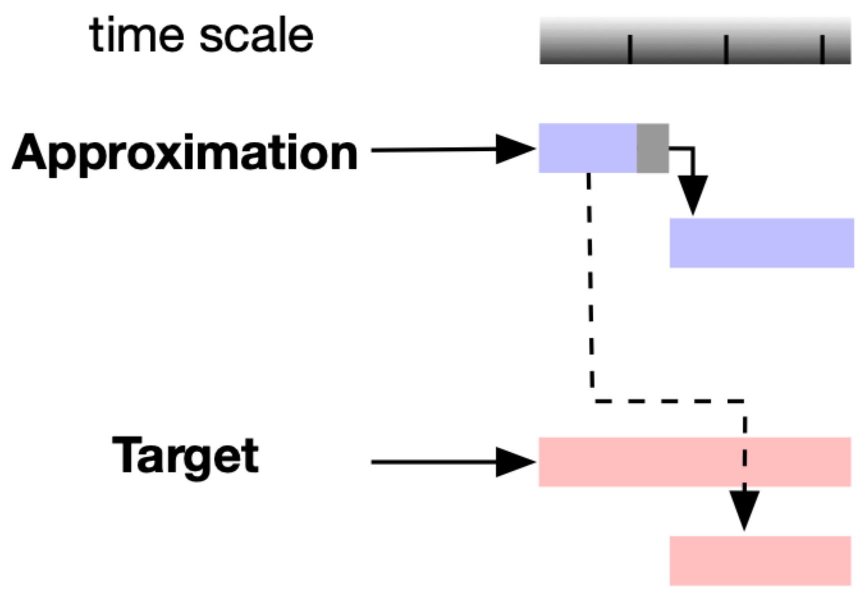
This image illustrates how time can be saved through speculative planning. With speculative planning, we do not need to wait for the target agent to complete each step before starting the next; instead, we only wait for the approximation agent to finish the step. Therefore, if the approximation agent computes some steps the same as the target agent, time can be saved.

In contrast, with normal agent planning, we must wait for the target agent to finish each step before proceeding to the next.

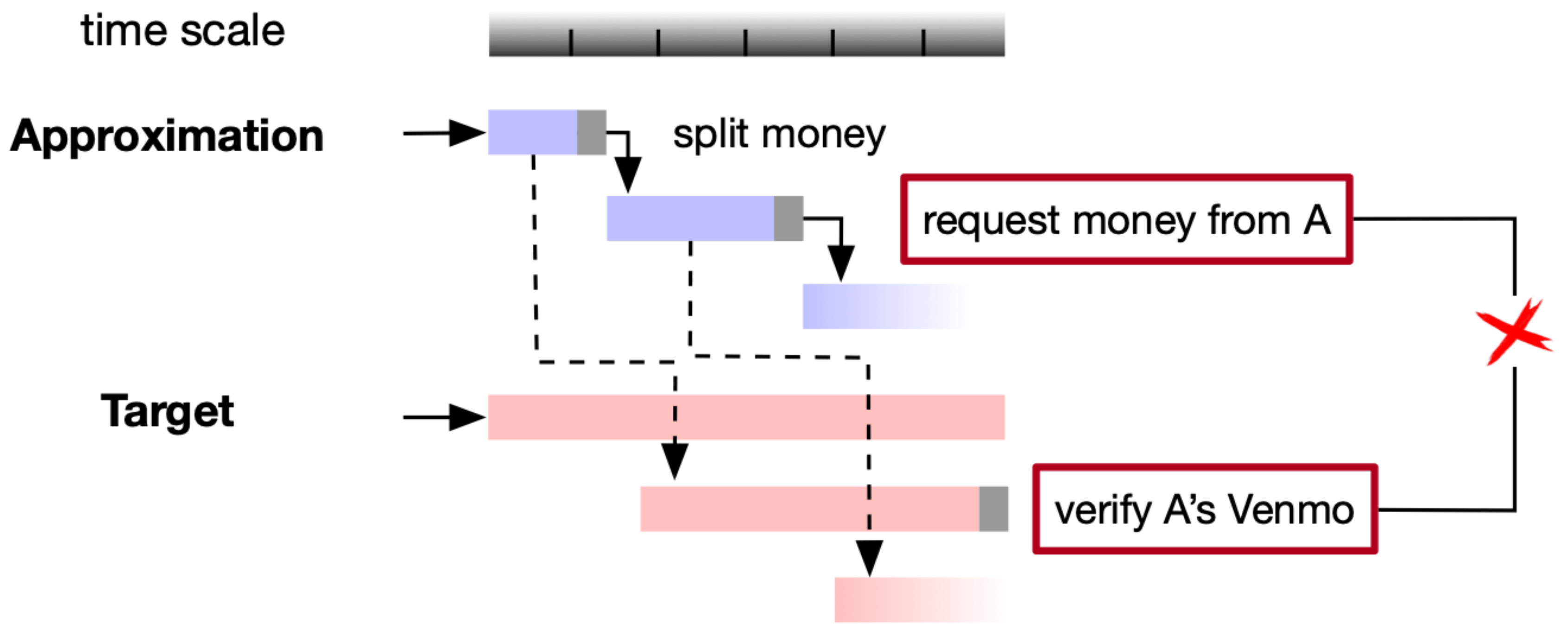
Speculative Planning Algorithm Broken-down



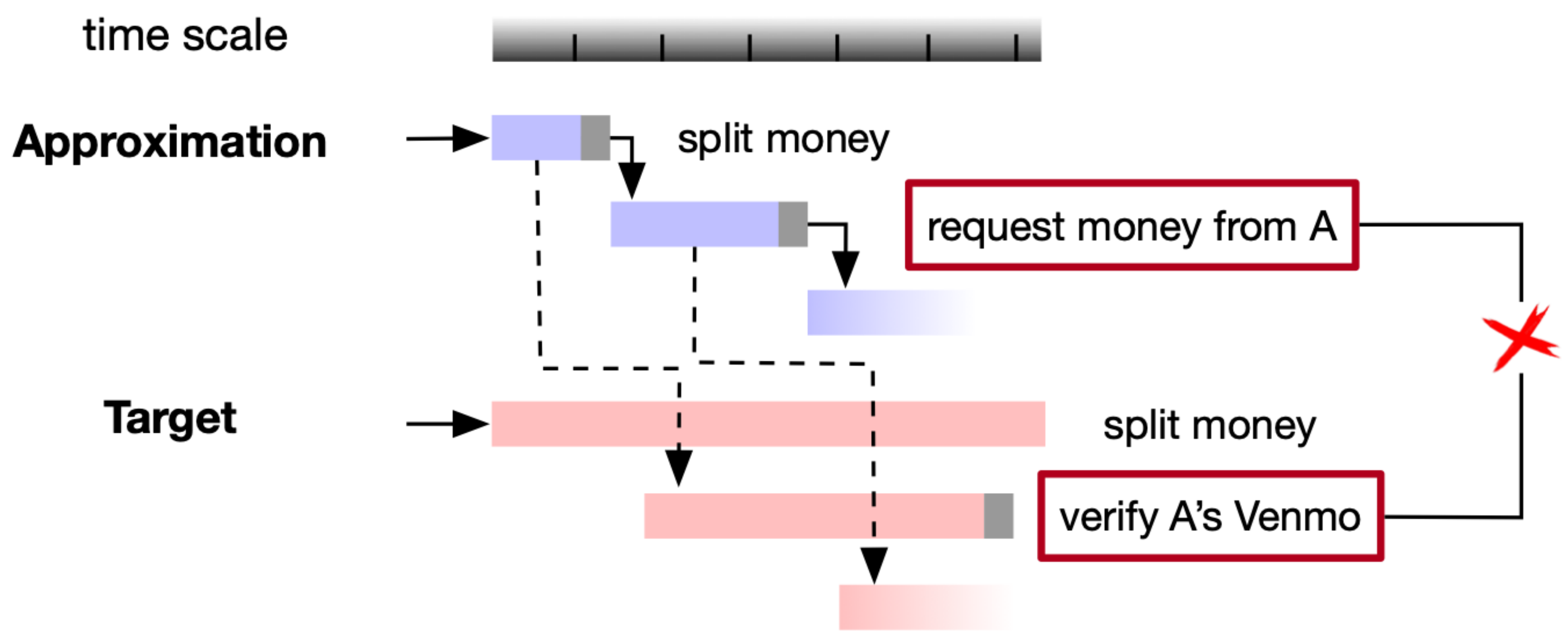
Speculative Planning Algorithm Broken-down



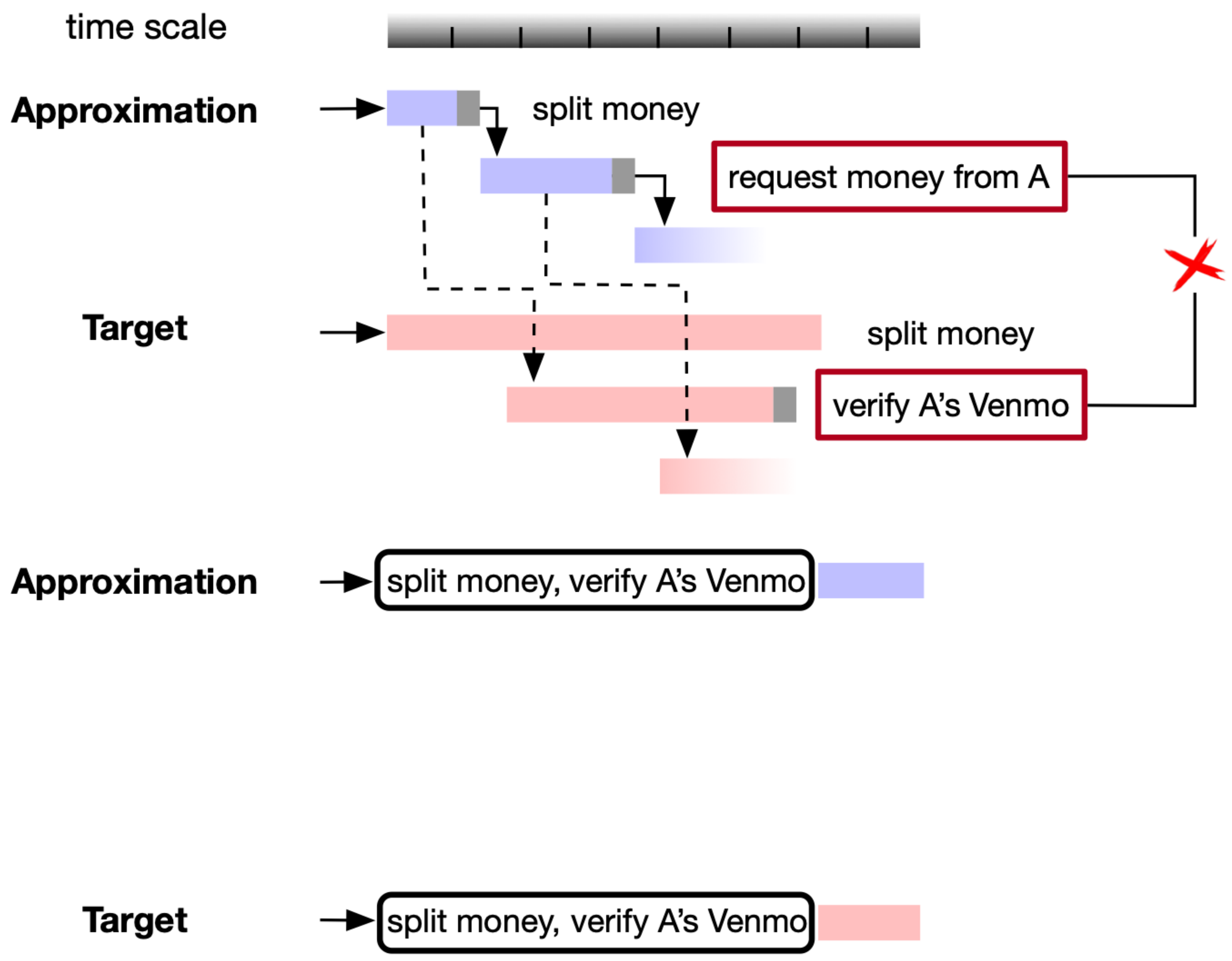
Speculative Planning Algorithm Broken-down



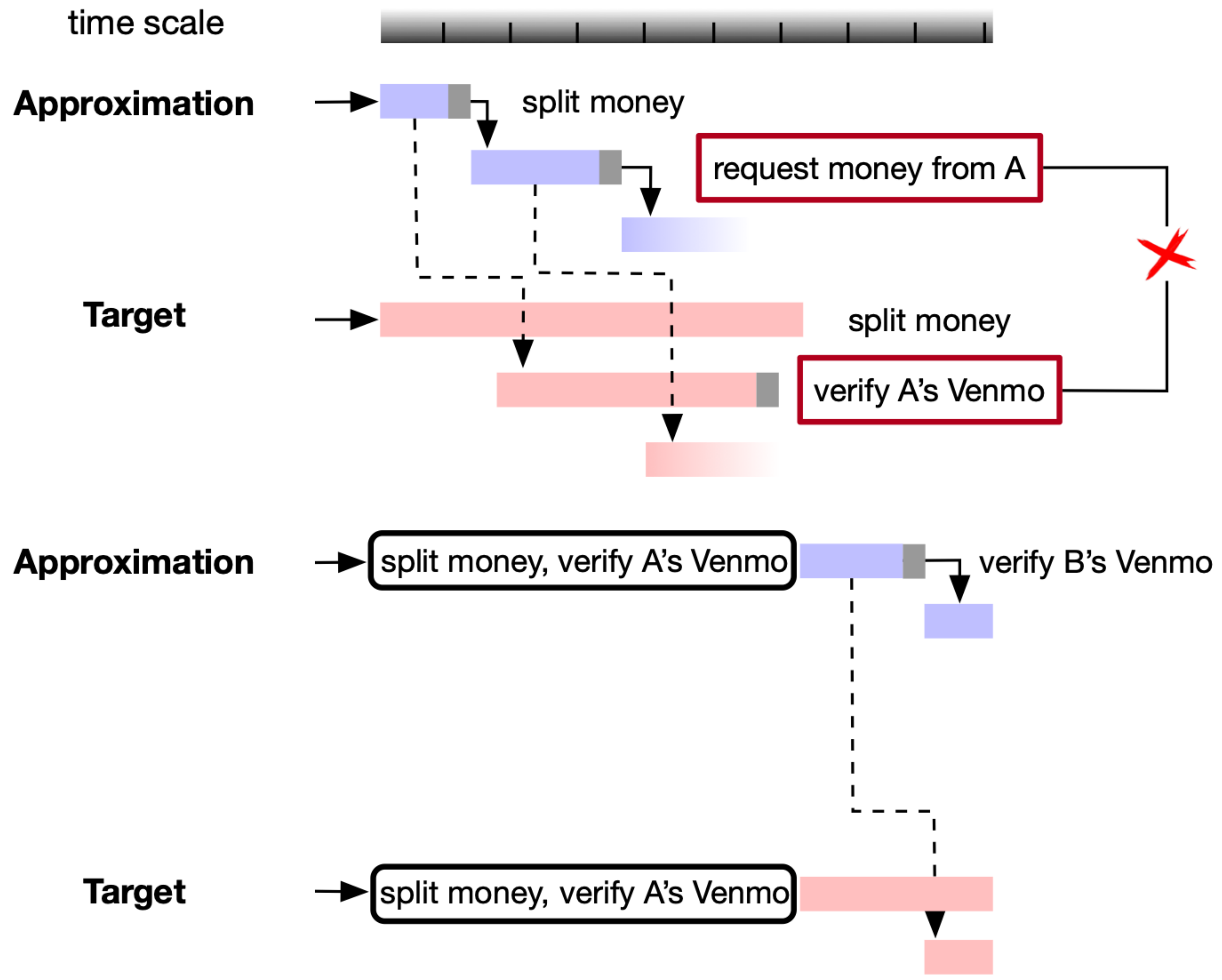
Speculative Planning Algorithm Broken-down



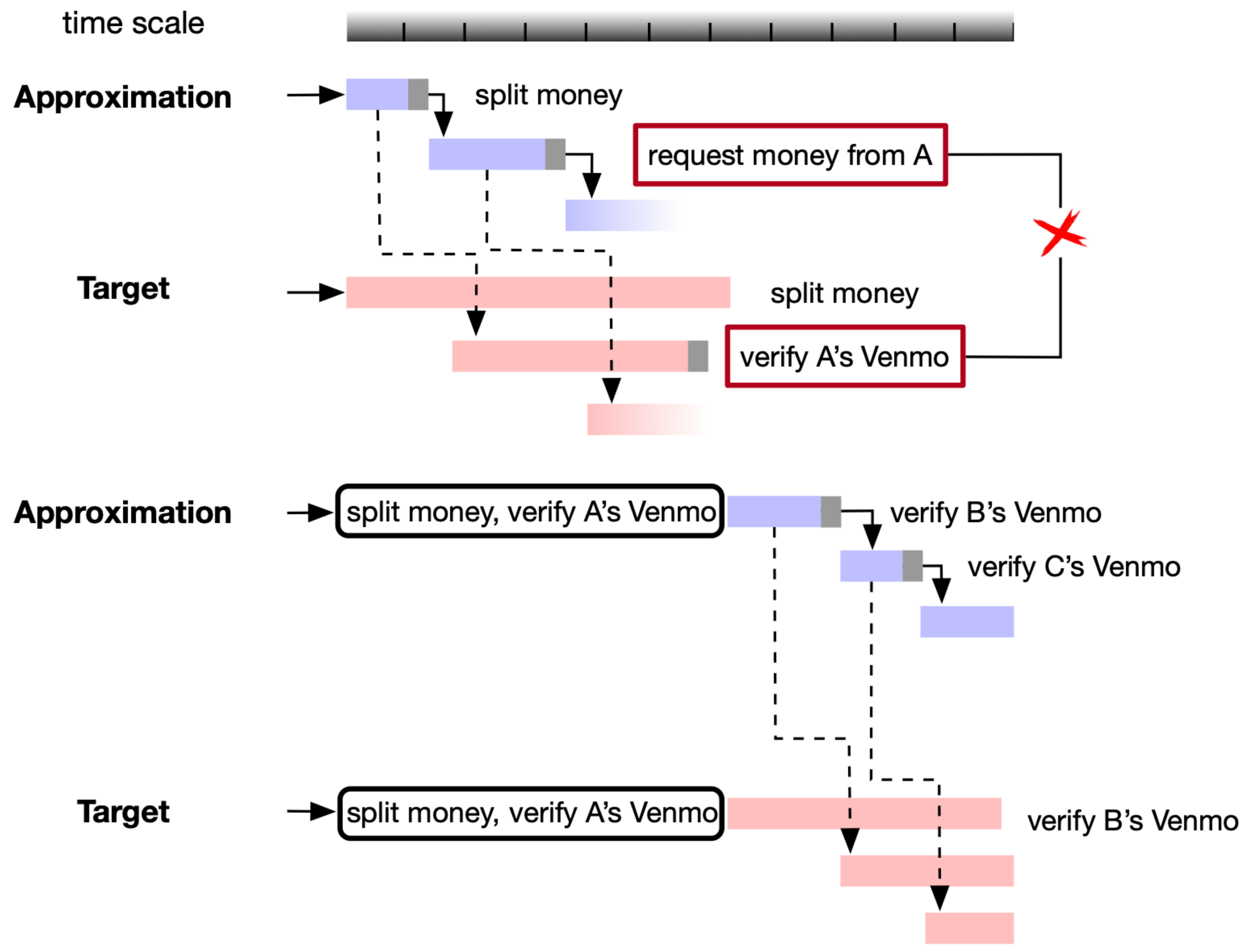
Speculative Planning Algorithm Broken-down



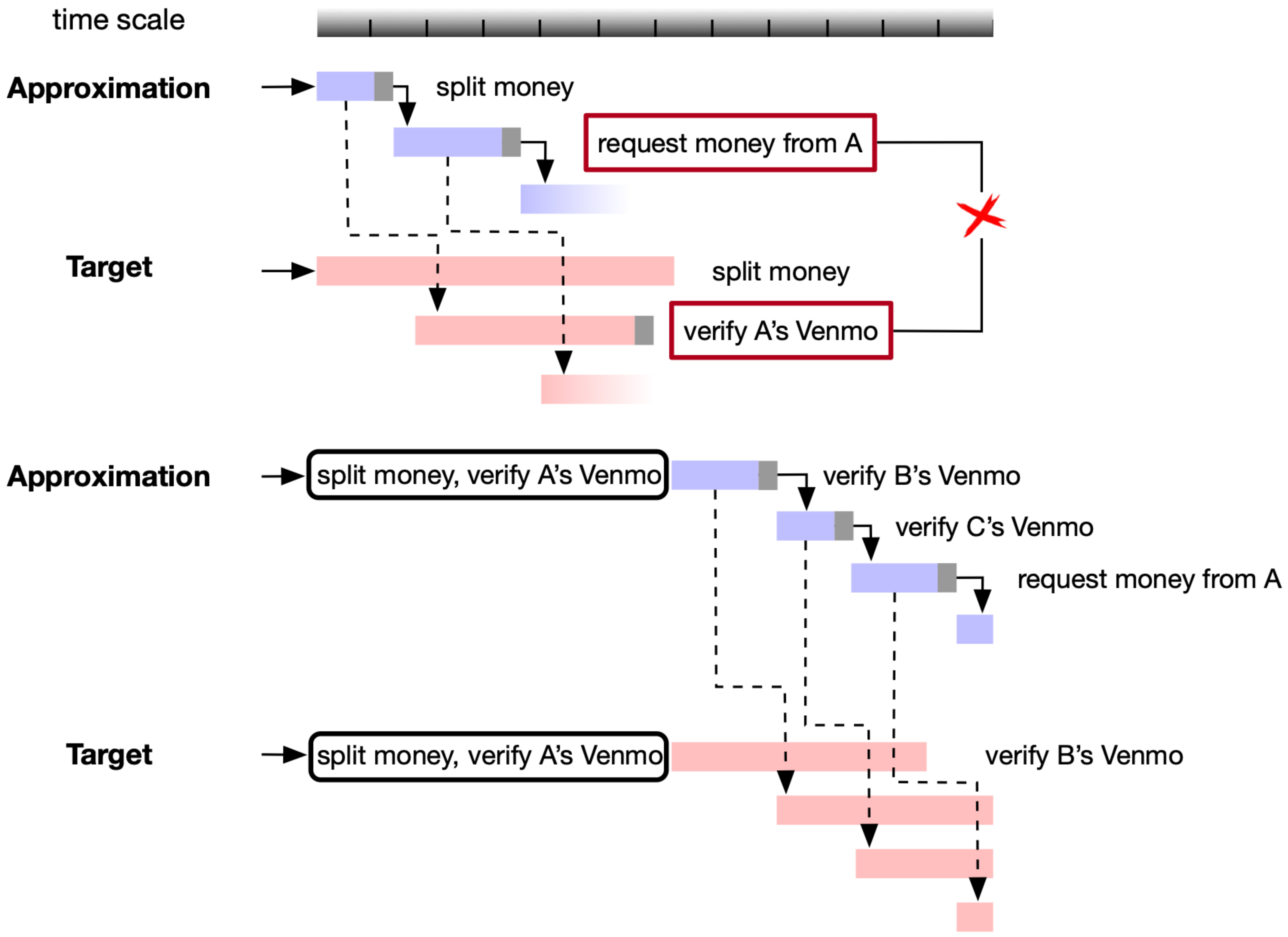
Speculative Planning Algorithm Broken-down



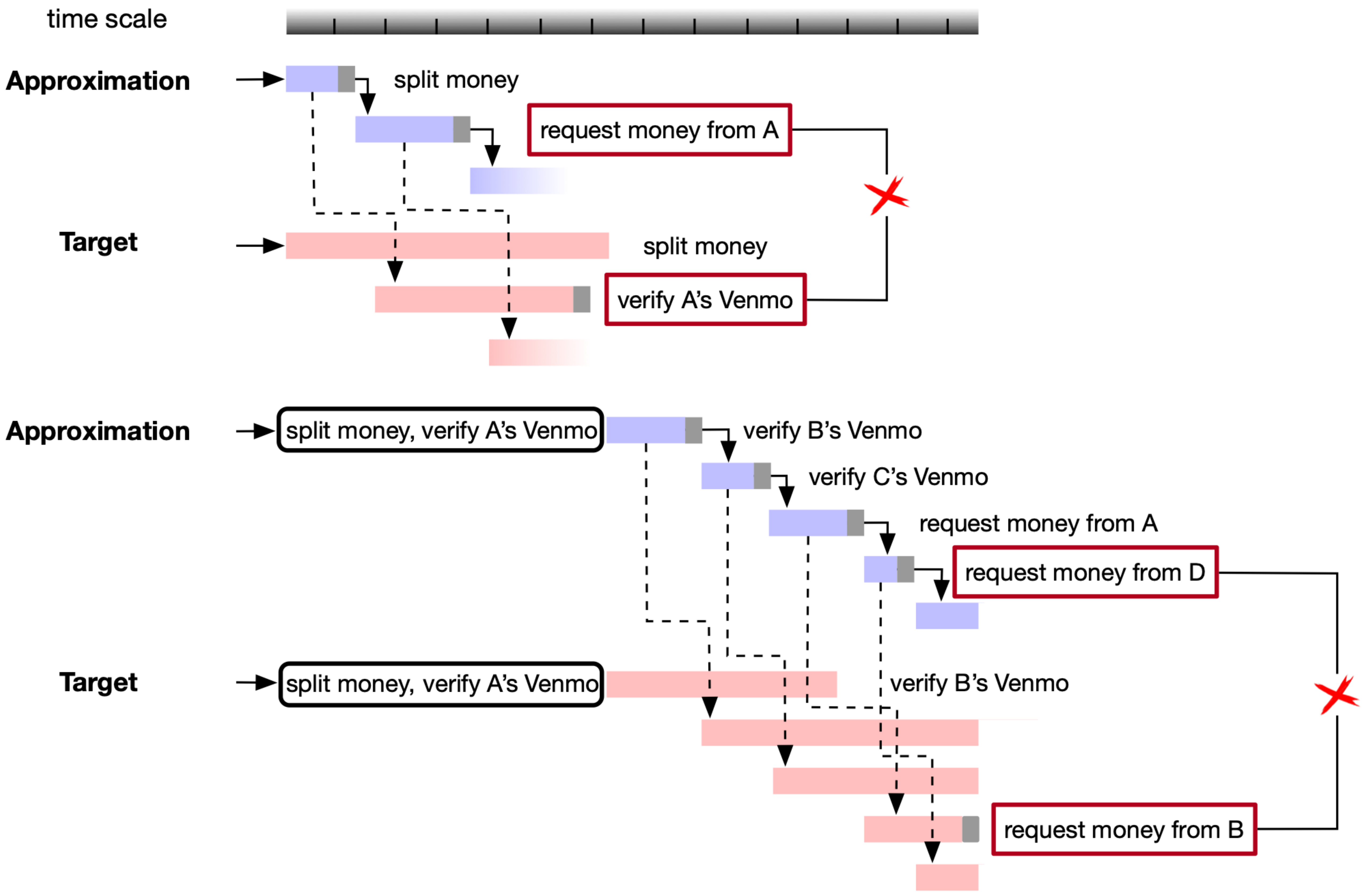
Speculative Planning Algorithm Broken-down



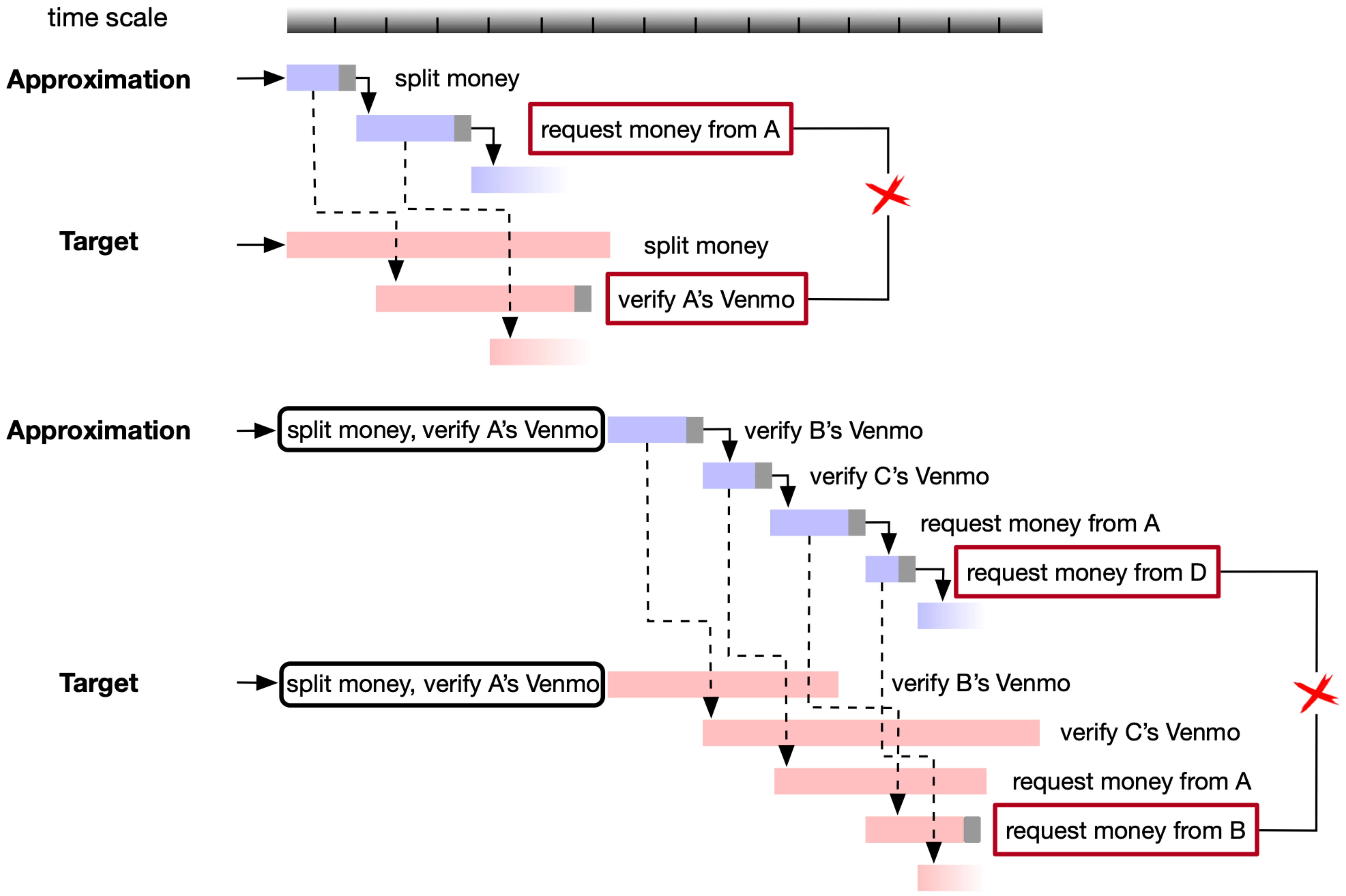
Speculative Planning Algorithm Broken-down



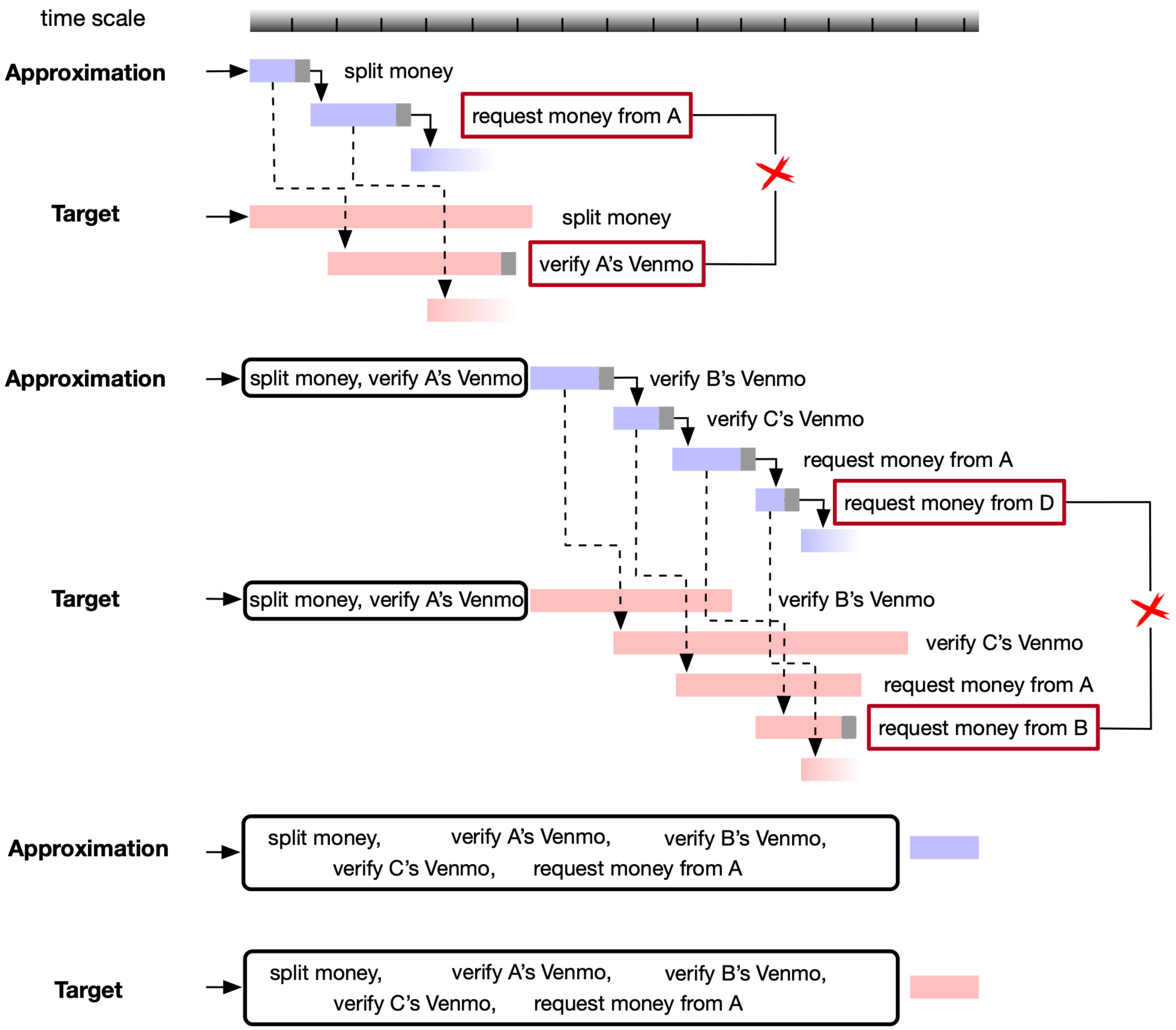
Speculative Planning Algorithm Broken-down



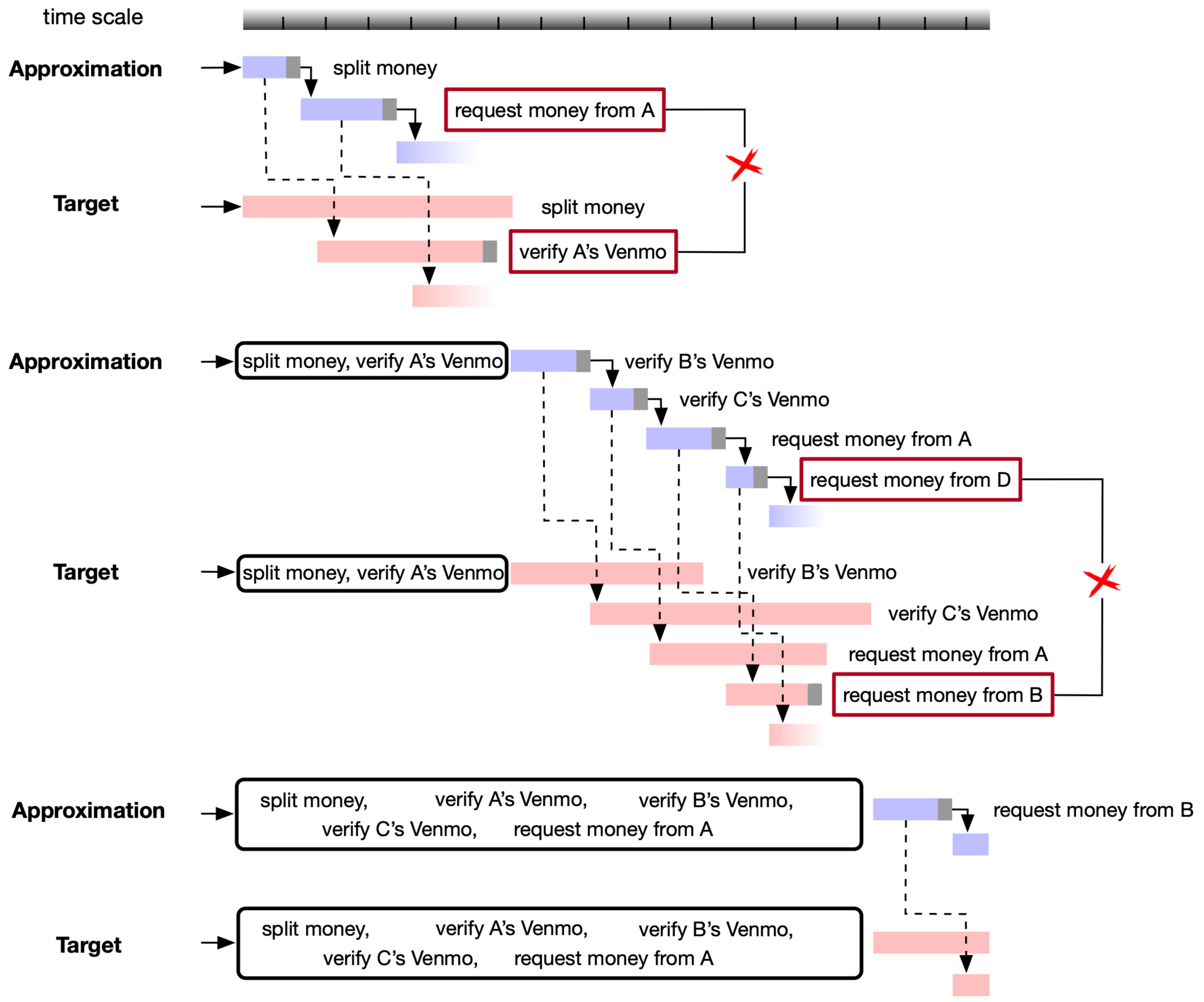
Speculative Planning Algorithm Broken-down



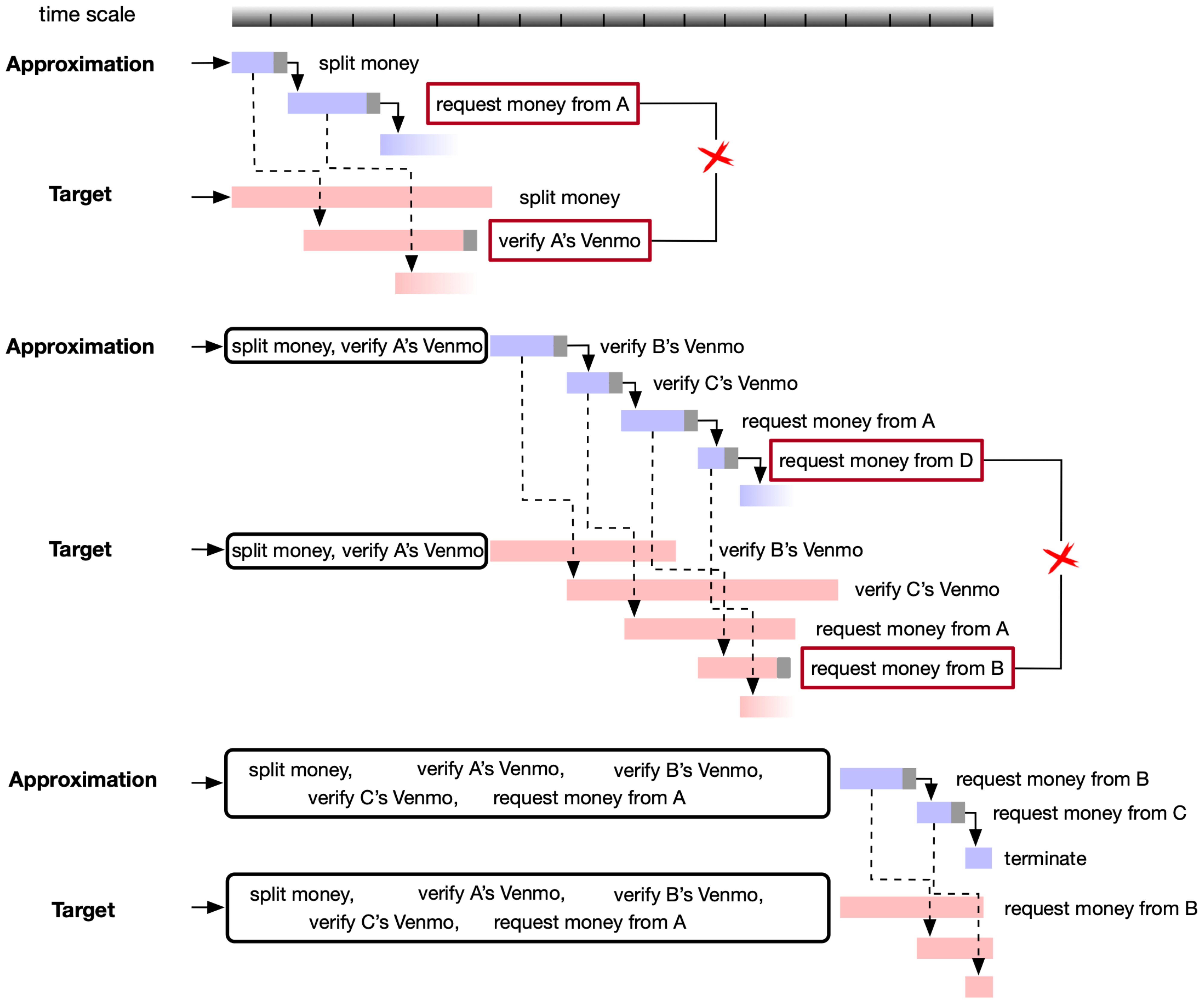
Speculative Planning Algorithm Broken-down



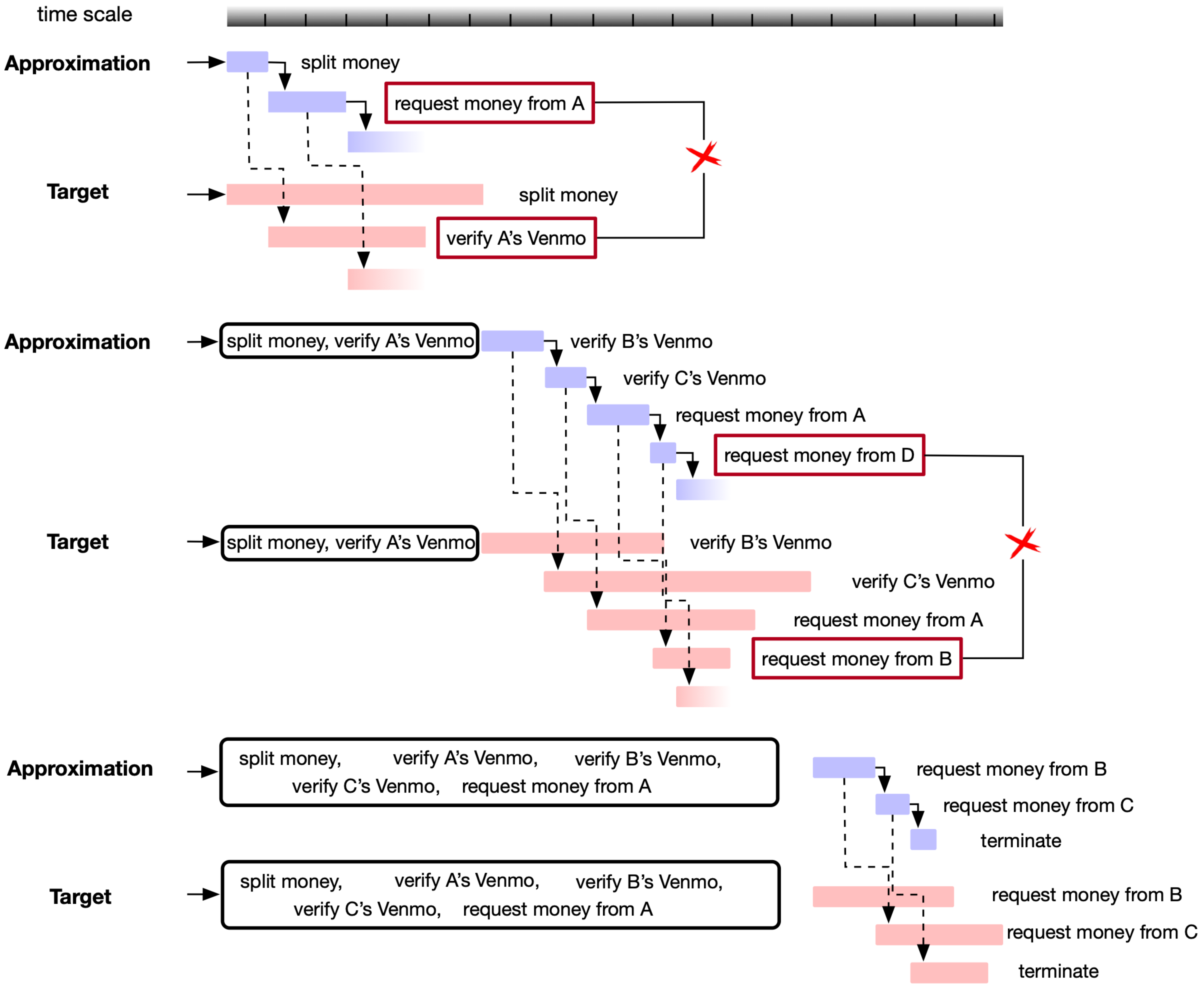
Speculative Planning Algorithm Broken-down



Speculative Planning Algorithm Broken-down



Speculative Planning Algorithm Broken-down



Simulation Experiment Result for Efficiency
Choice of approximation agent

(a) (b)

Figure (a): Faster approximation and more accurate approximation agent saves more time.
Figure (b): Speculative planning requires more tokens to be generated during the whole agent planning process. Assuming that approximation agent with different speed generates the same number of tokens: faster agent requires more tokens to be generated; more accurate agent requries fewer tokens to be generated.

Main Experiment Result

Setting 1: The approximation agent uses **direct-generation-based planning** with a GPT-4-turbo backbone, and the target agent uses **ReAct-based planning** with a GPT-4-turbo backbone.

Setting 2: The approximation agent uses **direct-generation-based planning** with a GPT-4-turbo backbone, and the target agent uses **chain-of-thought (CoT)-based planning** with a GPT-4-turbo backbone.

Setting 3: The approximation agent uses **CoT-based planning** with a GPT-4-turbo backbone, and the target agent uses **Multi-agent discussion planning** with a GPT-4-turbo backbone.

Setting 4: The approximation agent uses **direct-generation-based planning** with a GPT-3.5-turbo backbone, and the target agent uses **direct-generation-based planning** with a GPT-4-turbo backbone.

Main Experiment Result

Metrics	Setting 1	ReAct	Setting 2	CoT	Setting 3	MAD	Setting 4	DG
TT	$33.91_{\pm 30.38}$	$43.63_{\pm 25.39}$	$28.64_{\pm 25.49}$	$39.96_{\pm 27.25}$	$105.42_{\pm 50.84}$	$182.70_{\pm 421.49}$	$4.63_{\pm 1.78}$	$5.77_{\pm 1.83}$
Min-TT	6.80	9.16	3.53	8.60	28.24	50.89	1.70	2.23
ST	$5.92_{\pm 3.00}$	$8.69_{\pm 2.75}$	$5.52_{\pm 3.71}$	$7.98_{\pm 2.72}$	$21.50_{\pm 6.69}$	$34.84_{\pm 58.94}$	$1.14_{\pm 0.25}$	$1.49_{\pm 0.43}$
Min-ST	2.33	4.41	0.50	3.81	11.70	19.21	0.75	1.03
TO	$1920_{\pm 879.79}$	$1812.89_{\pm 832.30}$	$1770.61_{\pm 1010.44}$	$1397.90_{\pm 794.55}$	$6781.43_{\pm 3159.84}$	$4075.4_{\pm 1603.54}$	$107.05_{\pm 38.76}$	$40.13_{\pm 13.39}$
Min-TO	760	652	455	352	1754	1441	47	17
SO	$288.72_{\pm 65.29}$	$266_{\pm 44.37}$	$281.92_{\pm 88.77}$	$229.45_{\pm 44.23}$	$1385_{\pm 391.77}$	$836.65_{\pm 112.06}$	$26.47_{\pm 5.06}$	$10.14_{\pm 1.98}$
Min-SO	190.00	166.58	143.83	162.8	877	558.33	19.25	8.5
MC	$4.66_{\pm 0.59}$	$1_{\pm 0.00}$	$4.49_{\pm 0.82}$	$1_{\pm 0.00}$	$4.53_{\pm 0.56}$	$1_{\pm 0.00}$	$4.05_{\pm 0.21}$	$1_{\pm 0.00}$
Min-MC	3	1	3	1	3	1	4	1
cost	$\$0.122_{\pm 0.072}$	$\$0.0713_{\pm 0.026}$	$\$0.074_{\pm 0.040}$	$\$0.044_{\pm 0.018}$	$\$0.2973_{\pm 0.1387}$	$\$0.2160_{\pm 0.0795}$	$\$0.0012_{\pm 0.0011}$	$\$0.0012_{\pm 0.0004}$

Main Experiment Result on OpenAGI benchmark

On average, ~ 40% time cut

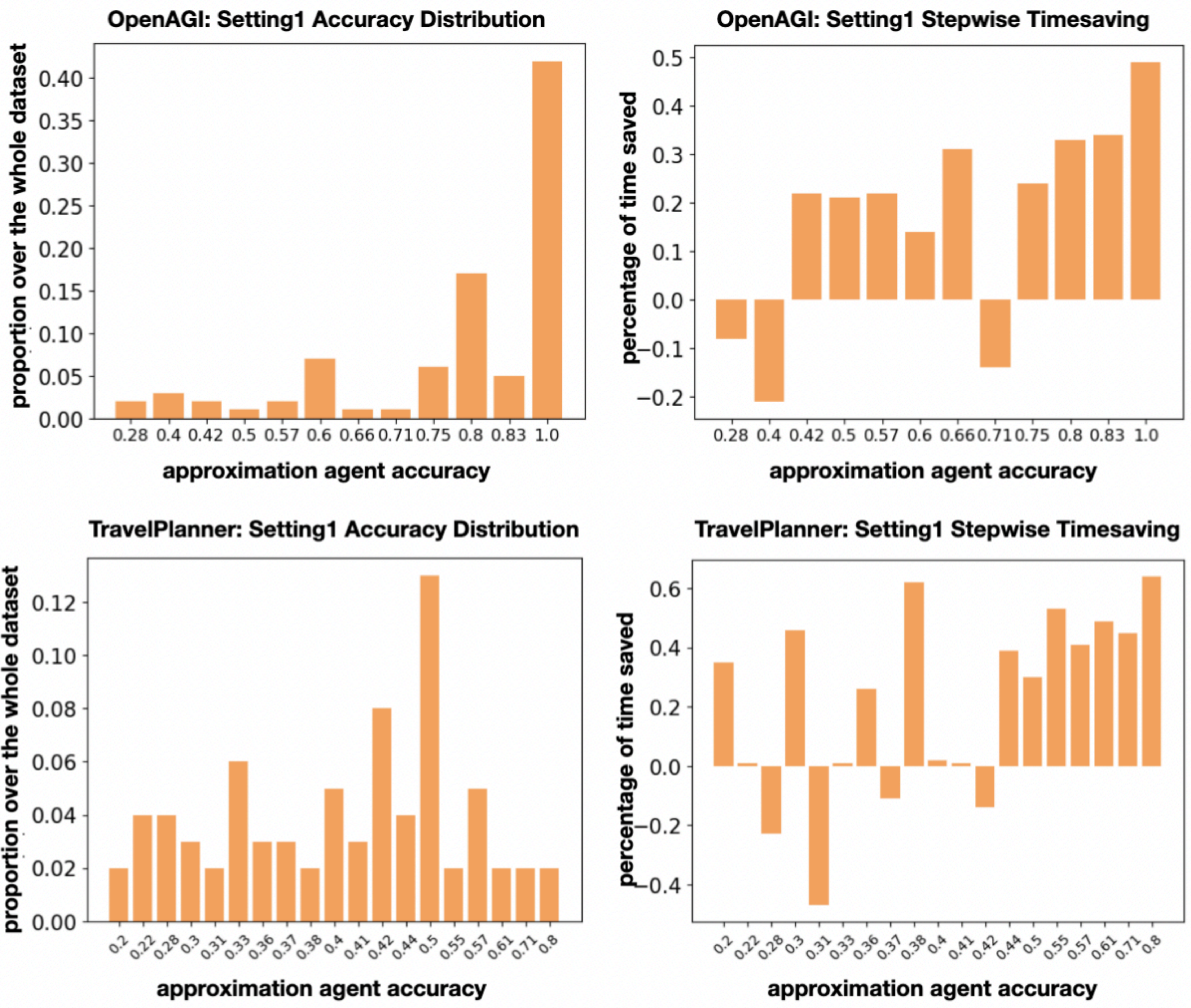
Main Experiment Result

Metrics	Settings							
	Setting 1	ReAct	Setting 2	CoT	Setting 3	MAD	Setting 4	DG
TT	$137.33_{\pm 66.39}$	$176.28_{\pm 77.18}$	$98.09_{\pm 45.02}$	$121.37_{\pm 32.18}$	$568.10_{\pm 292.99}$	$733.12_{\pm 290.51}$	-	-
Min-TT	40.78	55.18	29.22	42.09	149.00	127.59	-	-
ST	$11.16_{\pm 5.49}$	$14.13_{\pm 3.61}$	$10.71_{\pm 5.50}$	$12.75_{\pm 4.33}$	$27.53_{\pm 8.67}$	$40.03_{\pm 8.74}$	-	-
Min-ST	4.53	7.04	2.65	4.92	12.33	23.06	-	-
TO	$3751.94_{\pm 853.86}$	$2460.95_{\pm 332.07}$	$3082_{\pm 235.09}$	$2002.93_{\pm 276.54}$	$12353.84_{\pm 5872.86}$	$8976.39_{\pm 5371.31}$	-	-
Min-TO	1389	1762	833	1329	3443	2049	-	-
SO	$298.84_{\pm 128.97}$	$246.13_{\pm 56.34}$	$220.79_{\pm 56.19}$	$197.08_{\pm 87.68}$	$733.18_{\pm 477.72}$	$591.65_{\pm 467.82}$	-	-
Min-SO	128.13	108.30	85.42	68.06	189.00	186.27	-	-
MC	$5_{\pm 0.00}$	$1_{\pm 0.00}$	$5_{\pm 0.00}$	$1_{\pm 0.00}$	$5.00_{\pm 0.00}$	$1_{\pm 0.00}$	-	-
Min-MC	5	1	5	1	5	1	-	-
cost	$\$0.1583_{\pm 0.0367}$	$\$0.1038_{\pm 0.0033}$	$\$0.1393_{\pm 0.0241}$	$\$0.0874_{\pm 0.0125}$	$\$0.5941_{\pm 0.2871}$	$\$0.3990_{\pm 0.2309}$	-	-

Main Experiment Result on TravelPlanner benchmark

On average, ~ 30% time cut

Experiment Result Broken-down

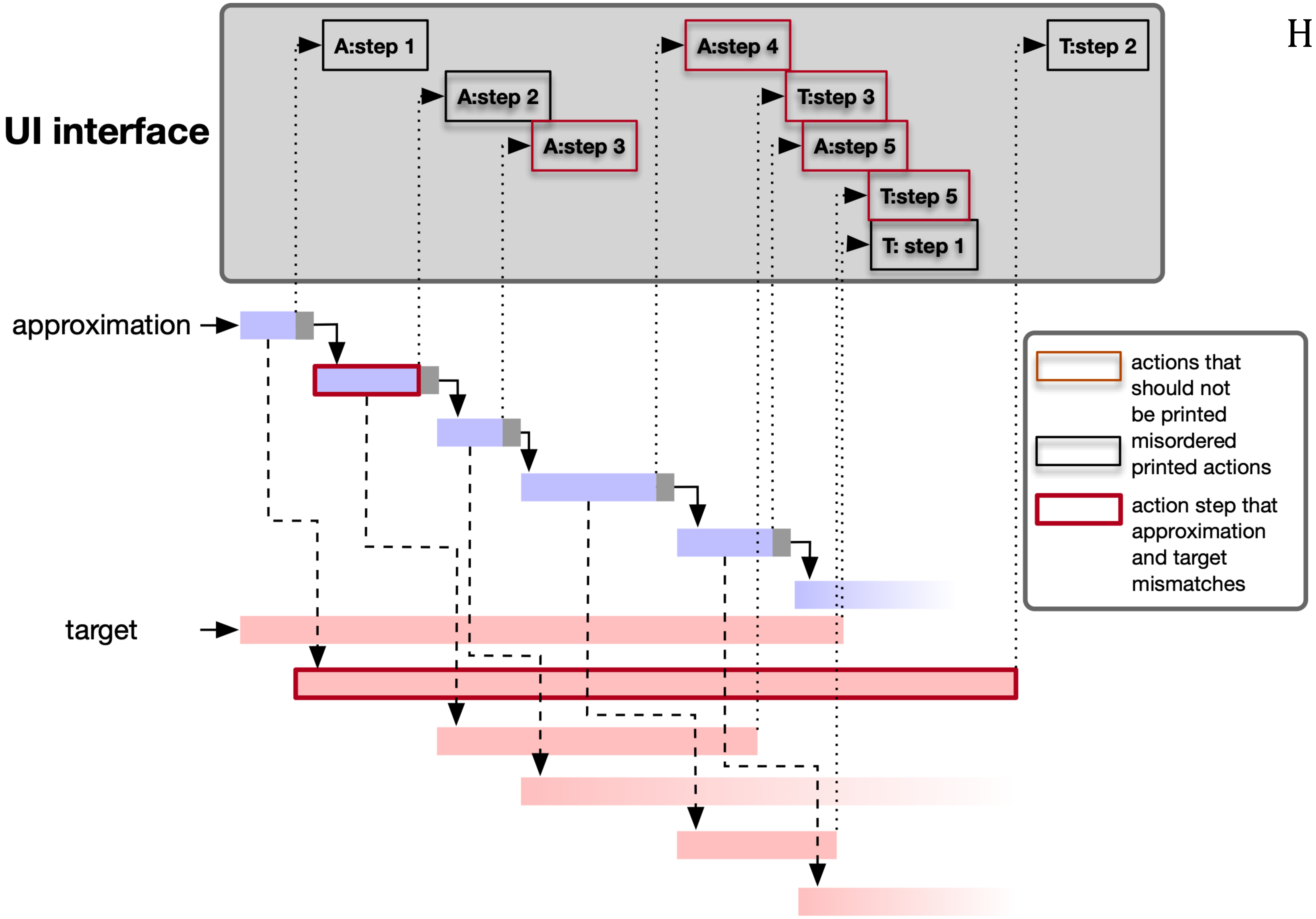


Sometimes negative results can be seen:

1. Concurrent requests may jam the computation resourec

2. Randomness in generation length and thus generation time

Interactive Speculative Planning: How to enable User Interaction?



Question 1:

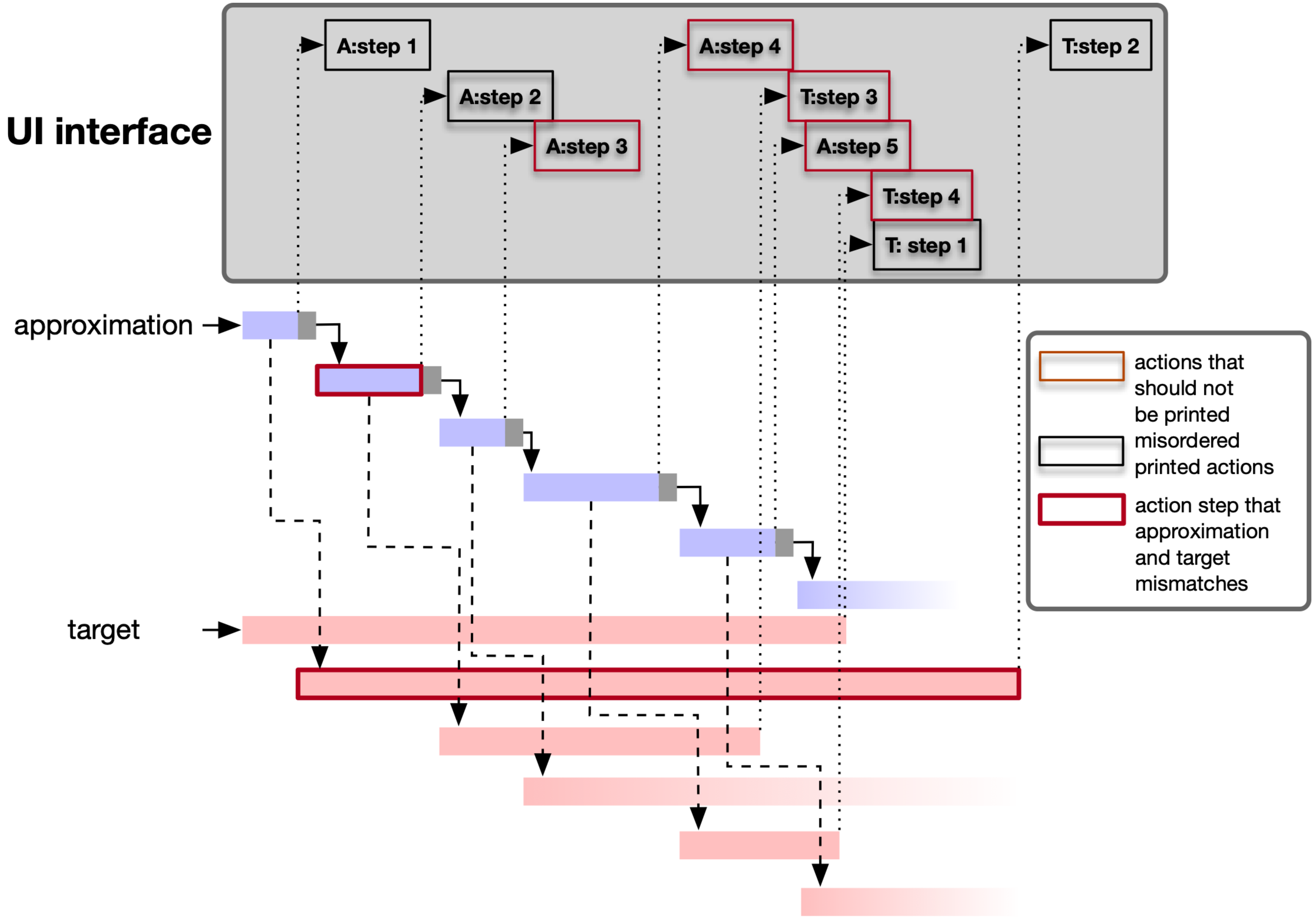
How to present system result to the UI?

It is important to note that immediately printing the outputs of the approximation agent and the target agent upon generation can be very confusing for two reasons:

(1) Issue 1: some outputs of the approximation agent should not be shown to the user at all

(2) Issue 2: the outputs of the target agent will not be sequential.

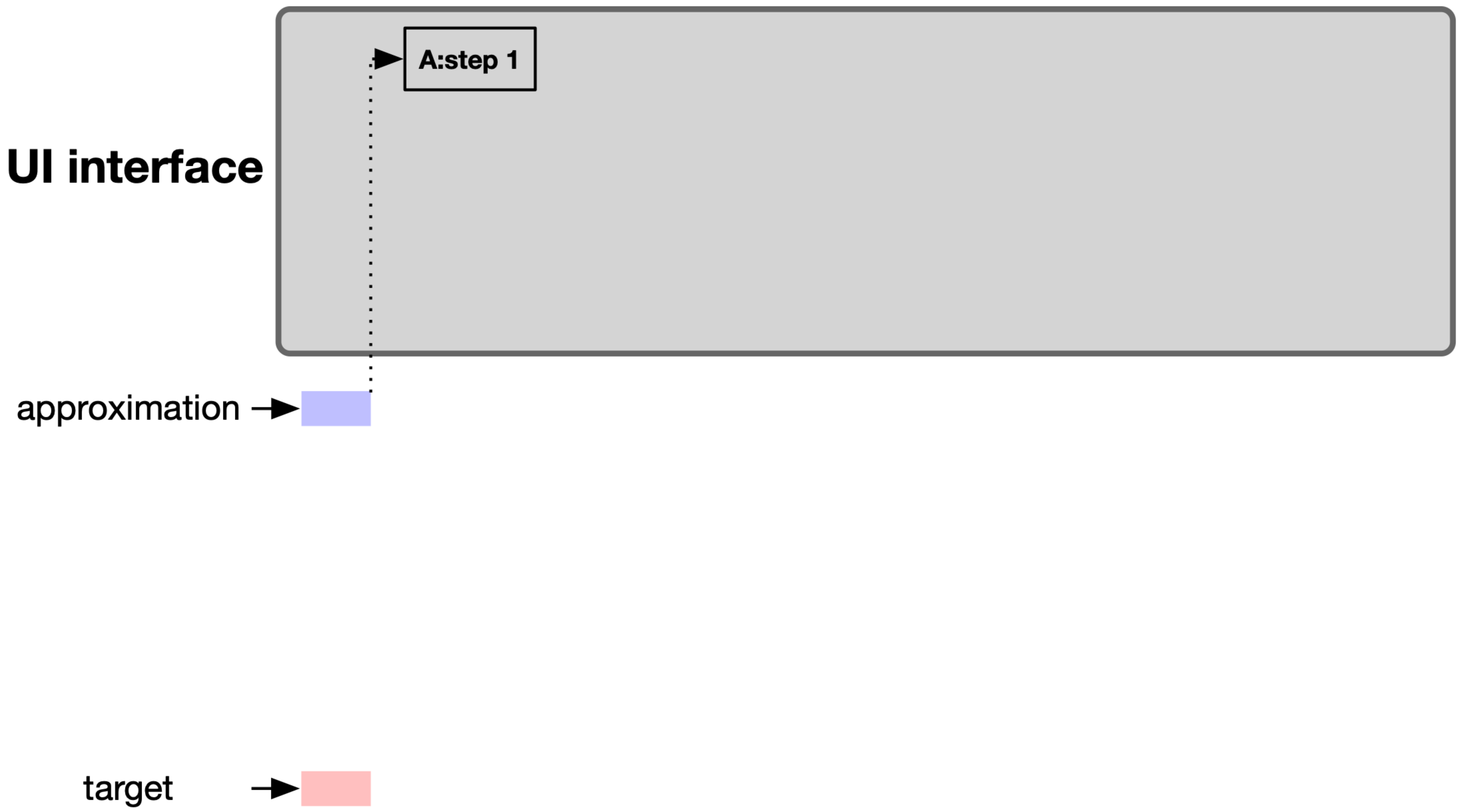
Interactive Speculative Planning:
How to present agent system running process to users?



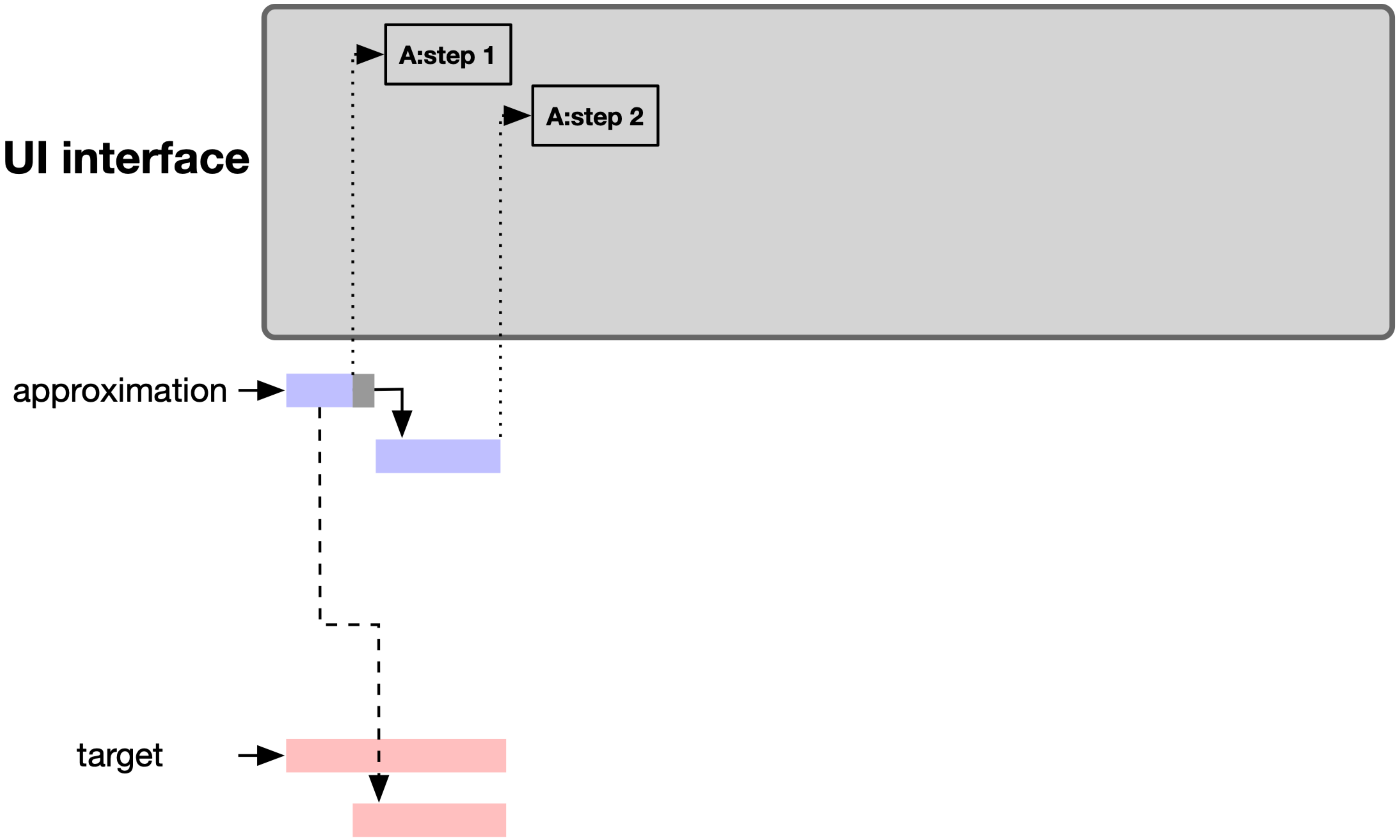
Issue 1: The approximation agent's output on the second step of the plan mismatches with the target agent's output, and *thus all results generated by the approximation agent based on the mistaken ``step 2'' will ultimately be discarded*. However, the immediate output of the agent's generation will print out the approximation's ``step 3, 4, 5,'' which are generated based on the wrong prefix, corresponding to the first issue.

Issue 2: As all target agent calls are asynchronous, the time each step is generated will *not follow any sequential order of the plan*, and thus the immediate printing out of the generated output will not be sequential either.

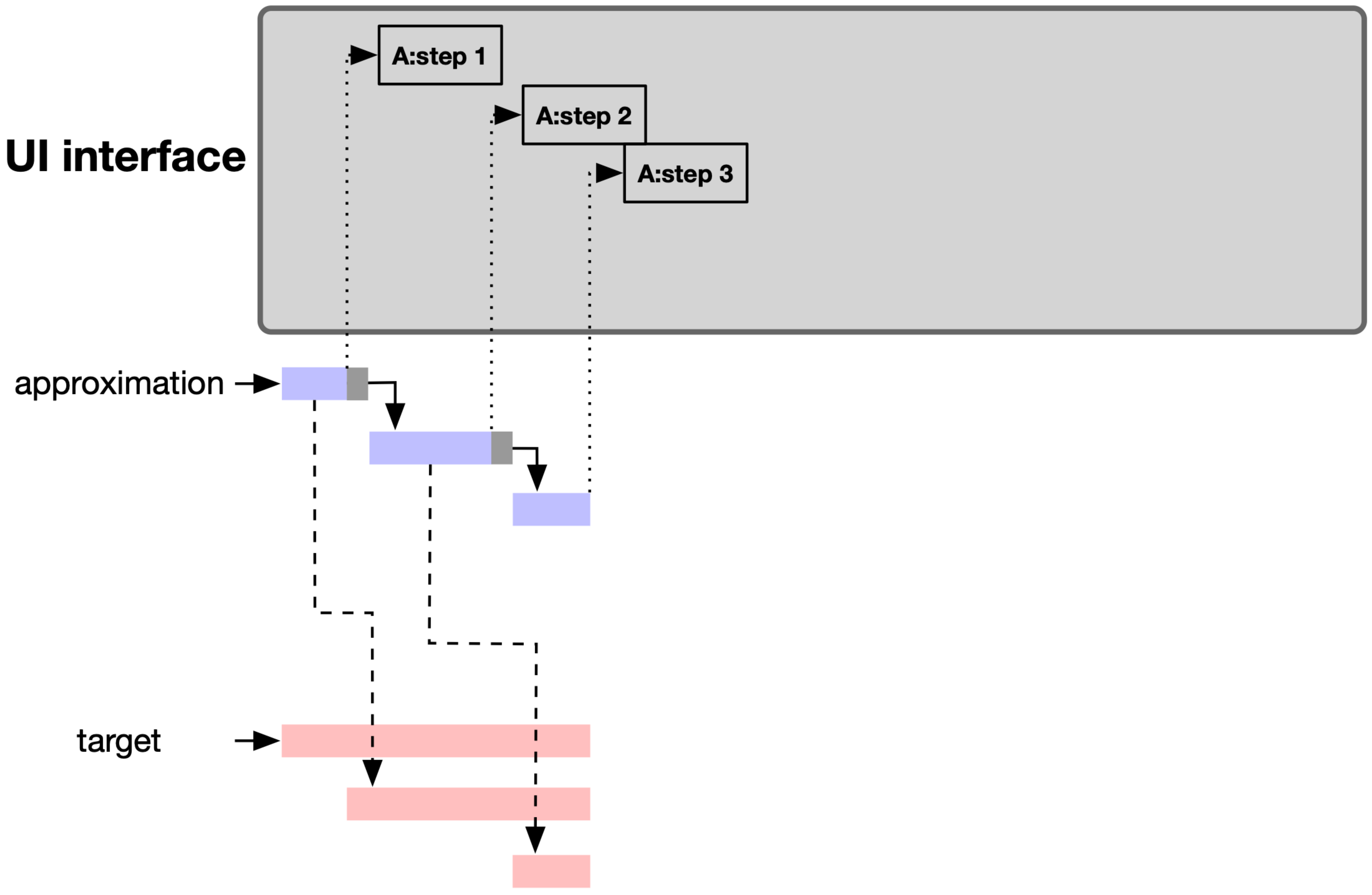
Immediate Presentation of Output is Problematic



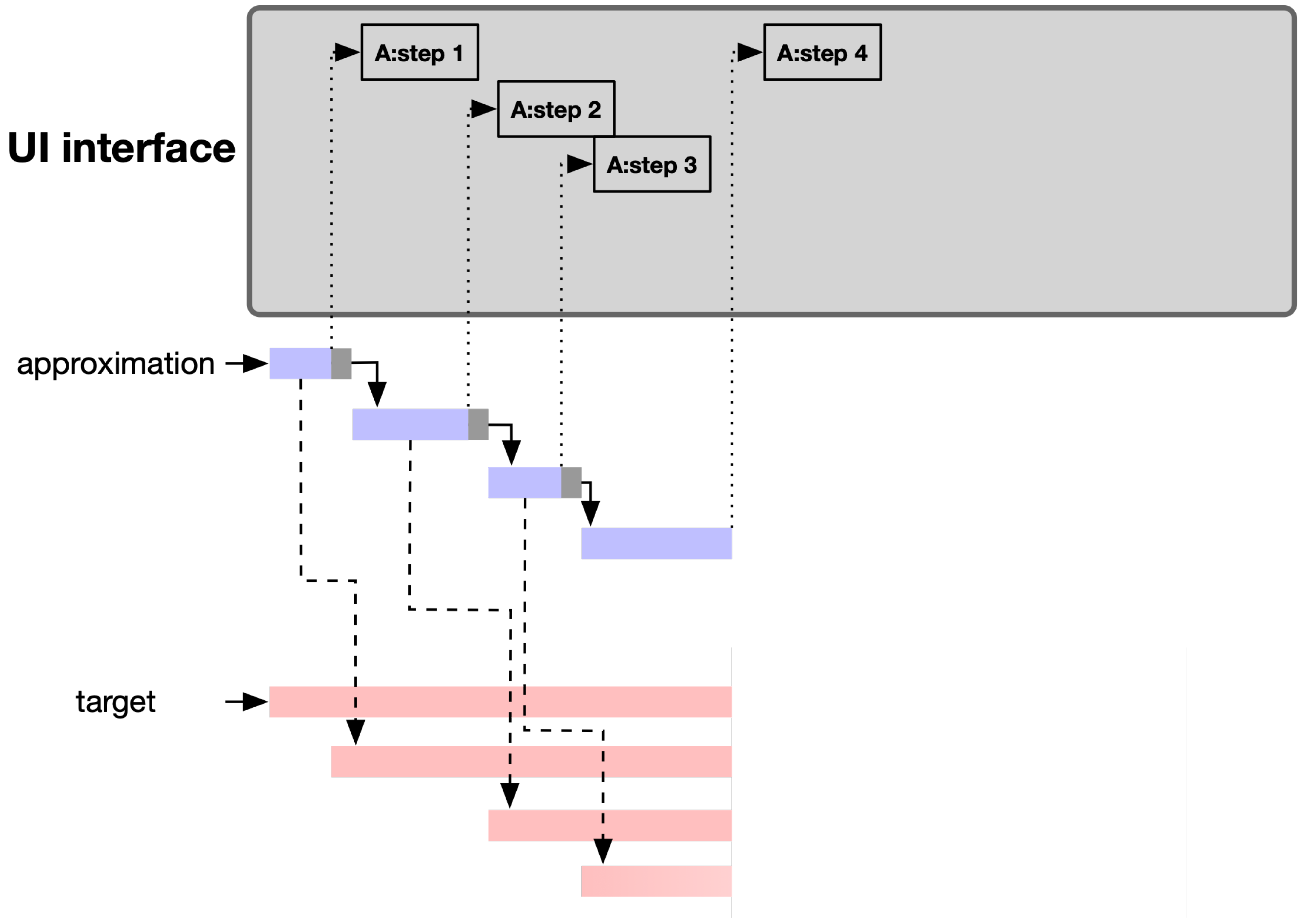
Immediate Presentation of Output is Problematic



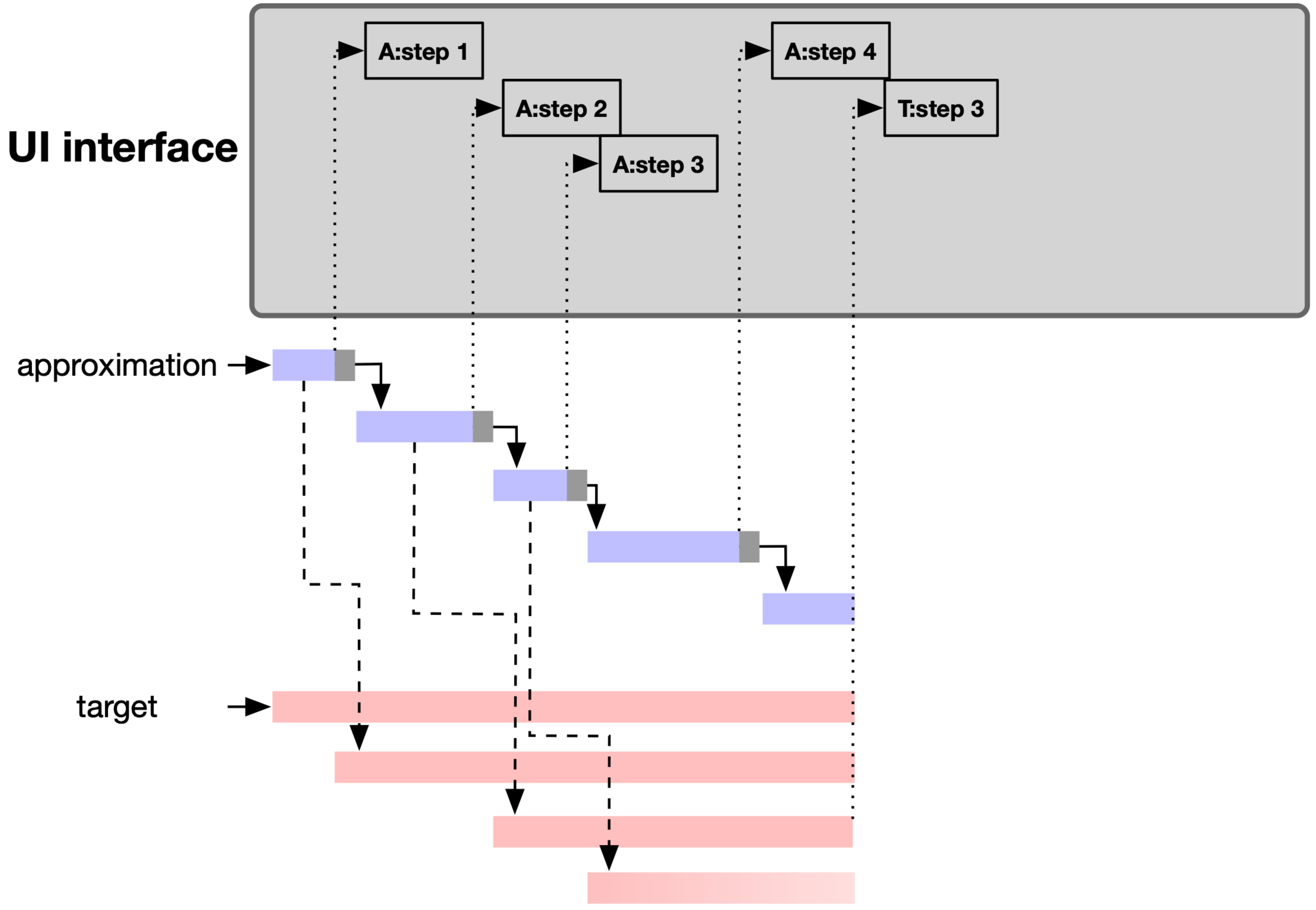
Immediate Presentation of Output is Problematic



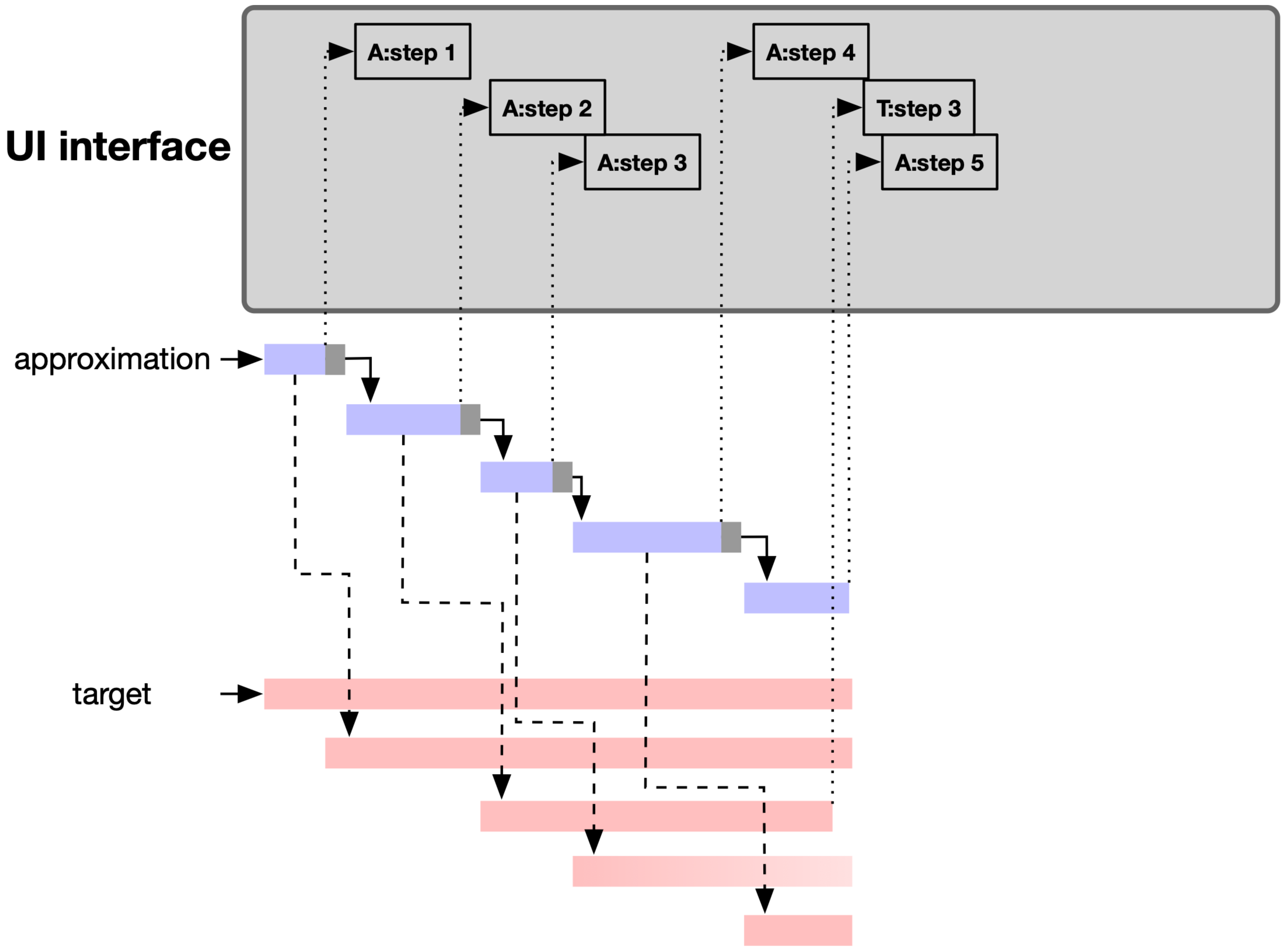
Immediate Presentation of Output is Problematic



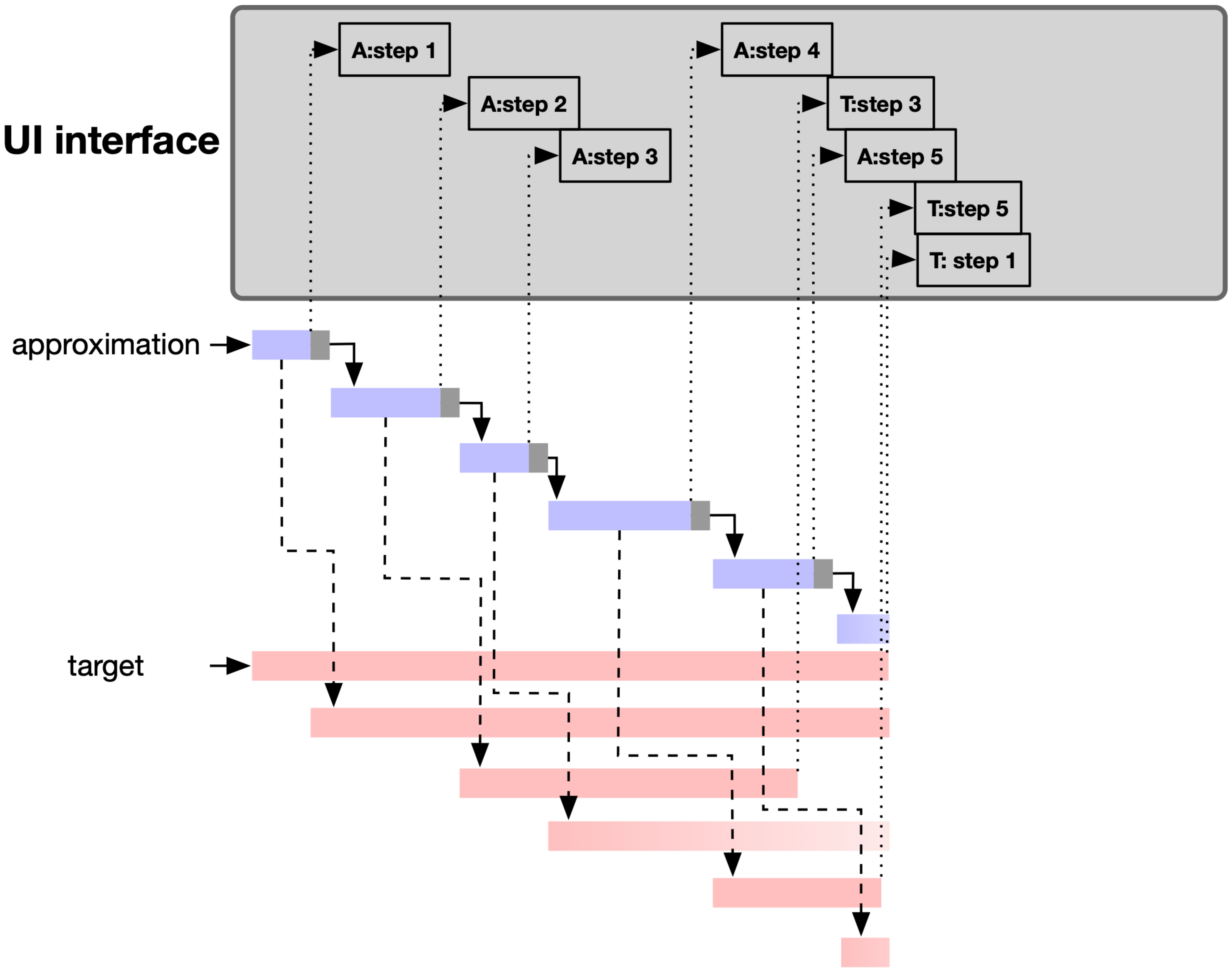
Immediate Presentation of Output is Problematic



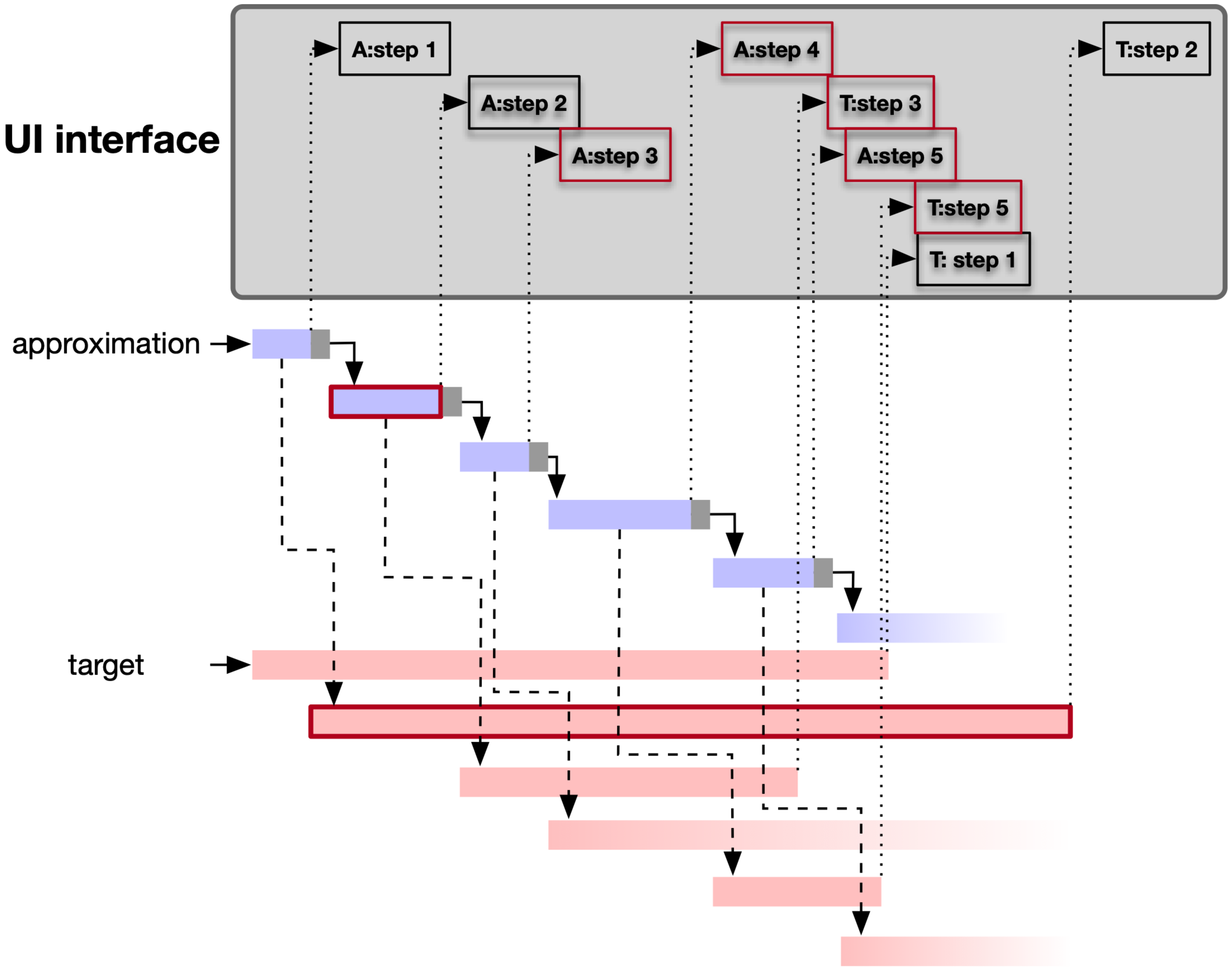
Immediate Presentation of Output is Problematic



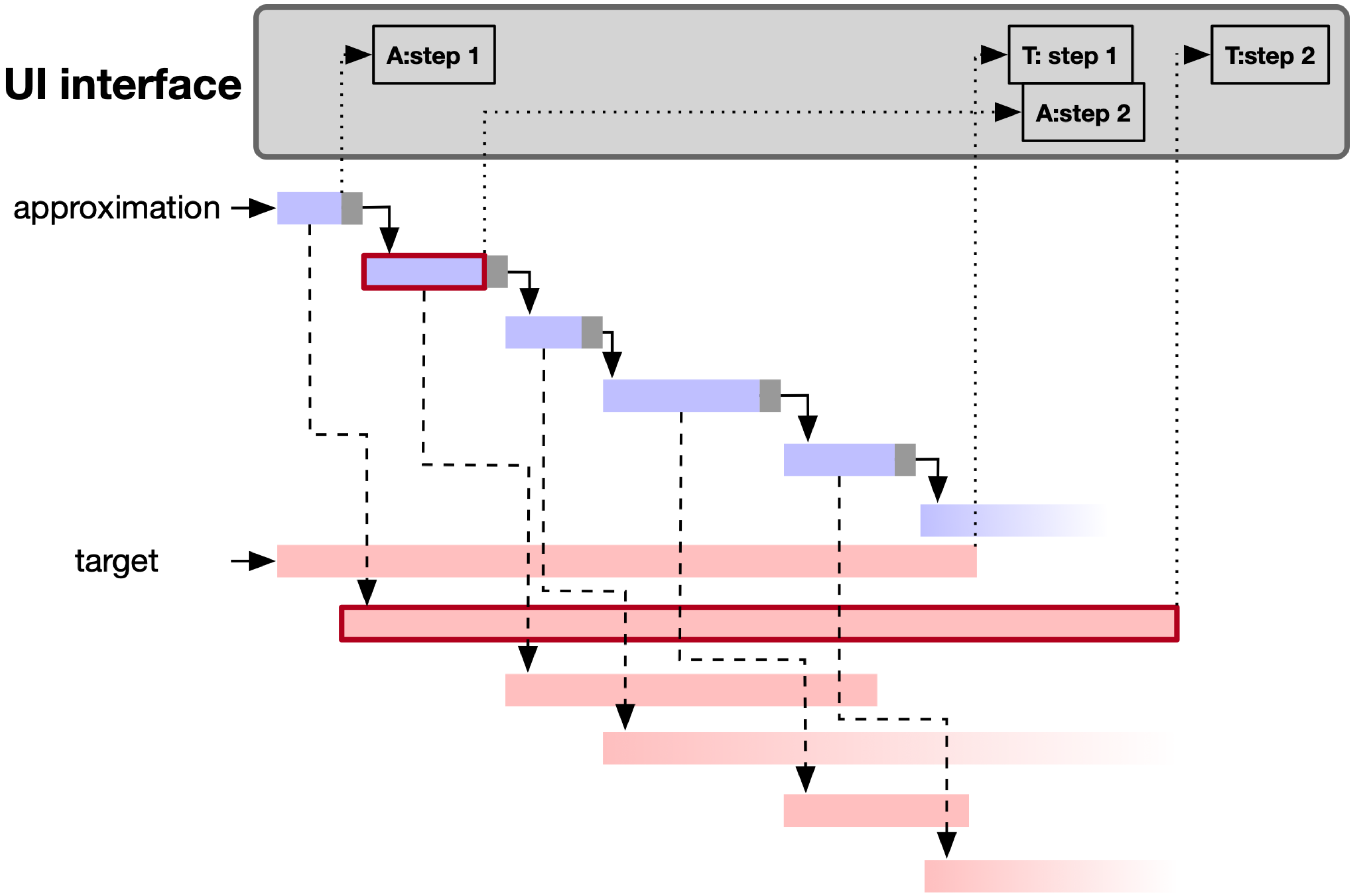
Immediate Presentation of Output is Problematic



Immediate Presentation of Output is Problematic



Interactive Speculative Planning: How to enable User Interaction?

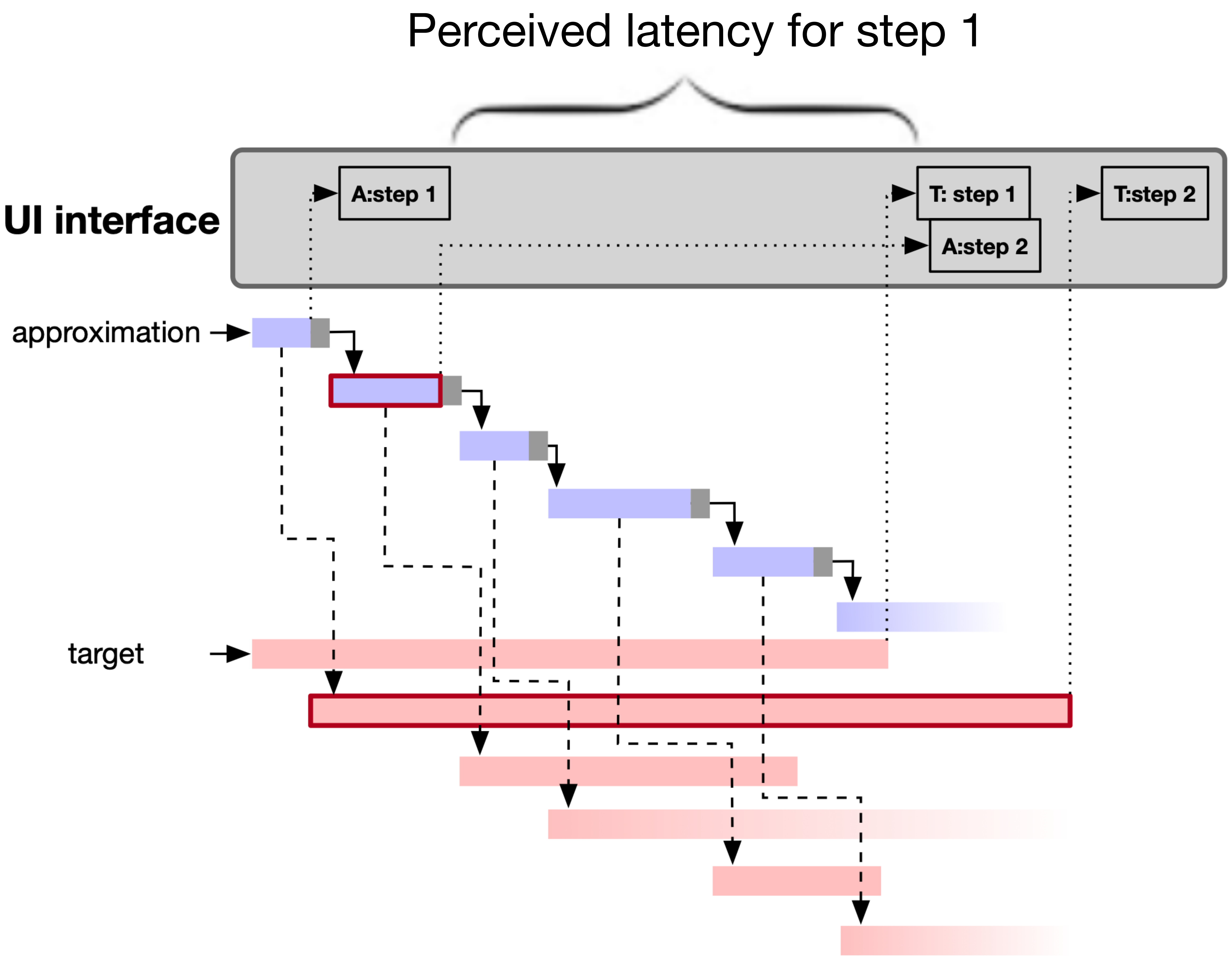


Question 1:
How to present system result to the UI?

Reschedule Mechanism:

The main idea is that for the approximation agent's output, the presents the i-th step from the approximation agent only after all preceding steps from the approximation agent have been confirmed to be consistent with the target agent, ensuring that **no consecutive unconfirmed steps are presented**.

Interactive Speculative Planning: How to enable User Interaction?

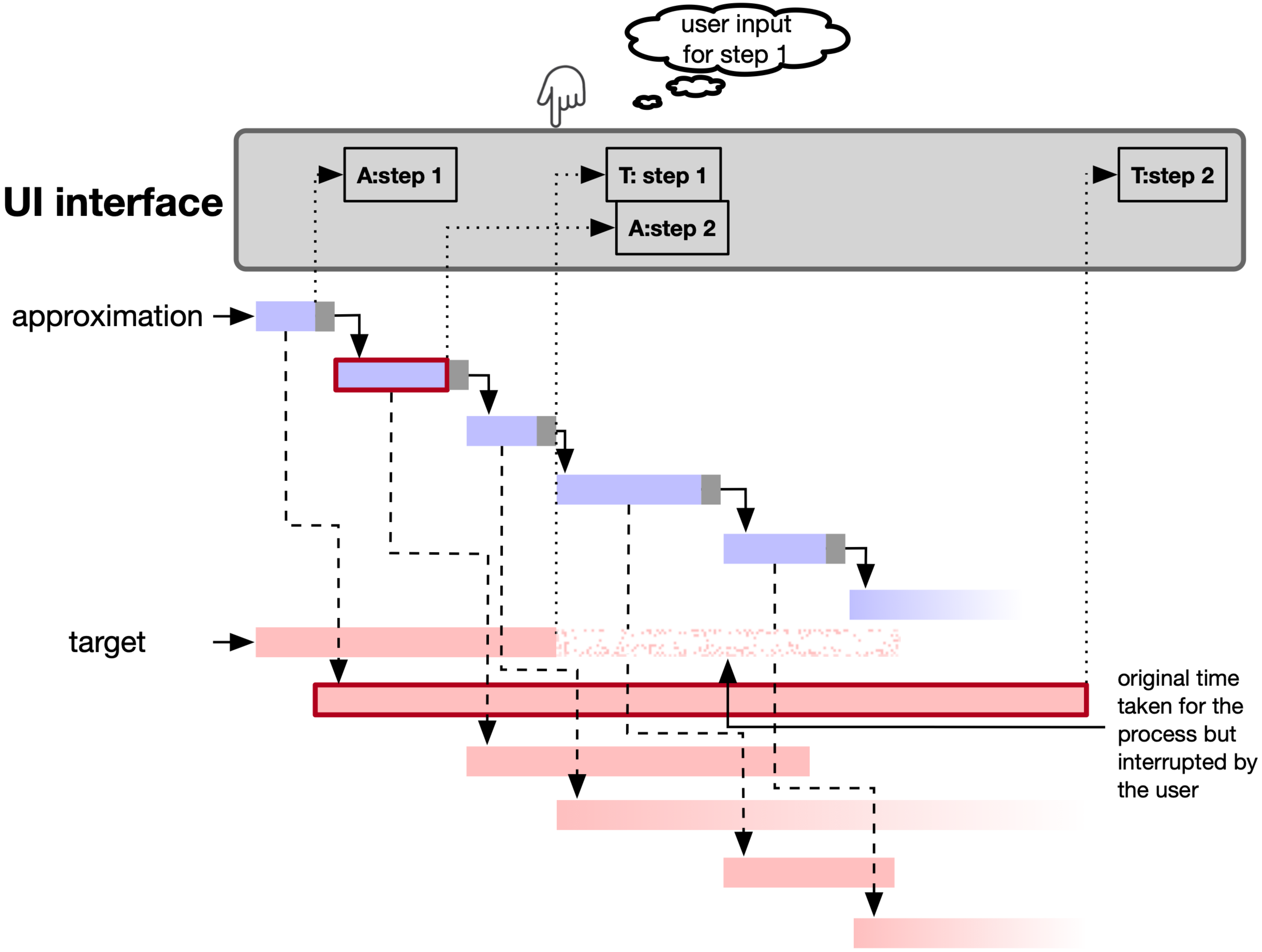


Question 2:
When do we expect user to interrupt?

Since the UI interface presentation for the i-th step of the plan can indicate the latency between the presentation of the approximation output A_i and the target output T_i, users can choose to interrupt during the observed latency and input their own value.

Therefore, user interaction here can again accelerate the whole system's speed.

Interactive Speculative Planning: How to handle user interruption?



Interactive Speculative Planning Summary

➢ How to accelerate agent planning by system design? Speculative Planning

➡ A fast approximation agent + an accurate/slow target agent

➢ How to enable active user interaction and leverage users to further accelerate agent planning? Latency-specific interaction: Interactive Speculative Planning

➡ Speculative planning system + latency-specific user interaction

⬇

A tri-agent acceleration system: approximation agent, a target agent, a human “agent”

Thank you!