

Diagnosing
Instability

Controlling
Behavior

Budgeting
Cost

Challenge 3: Token Economy: Budget



Ask for more budget



Switch tasks / hand off tasks

Budget-Aware Agent Learning



What's Left for Humans?

Complexity Inflation: OPENCLAW

~1M 16K+

Lines of Code

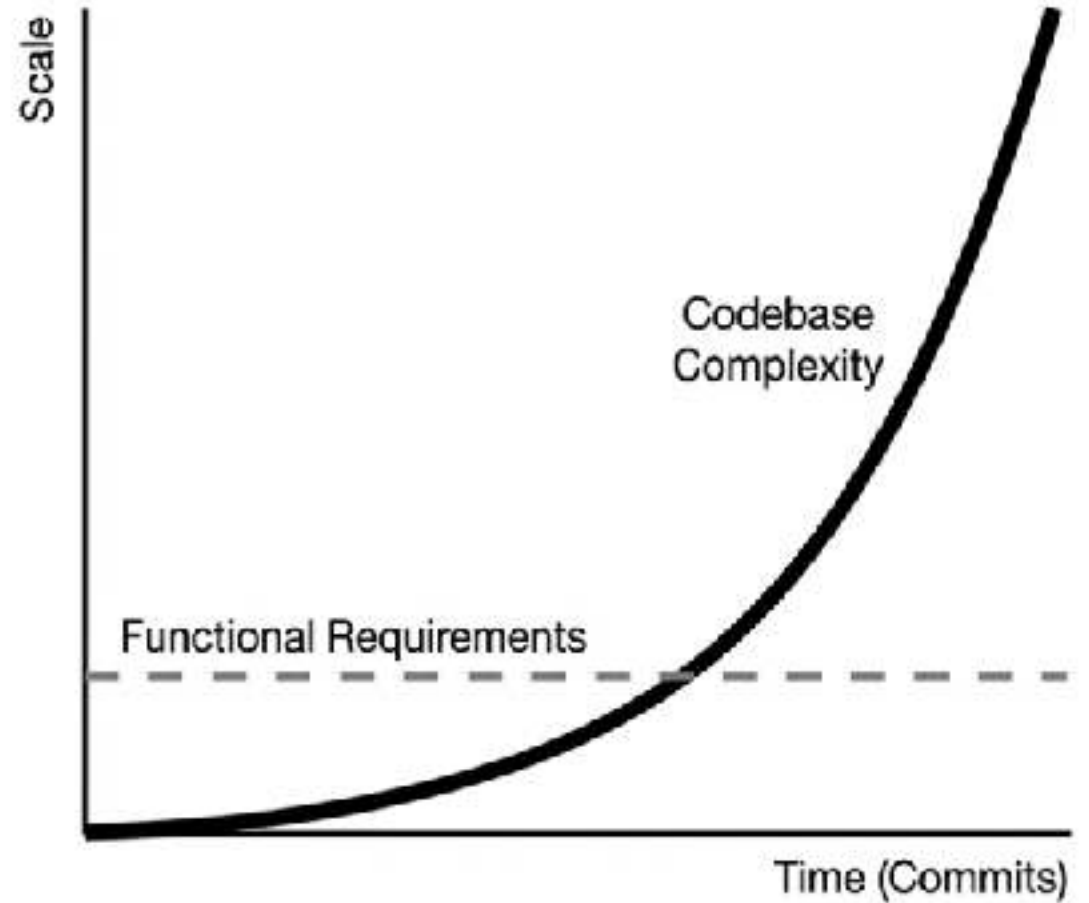
Commits

Ugly, Not Manageable

~400

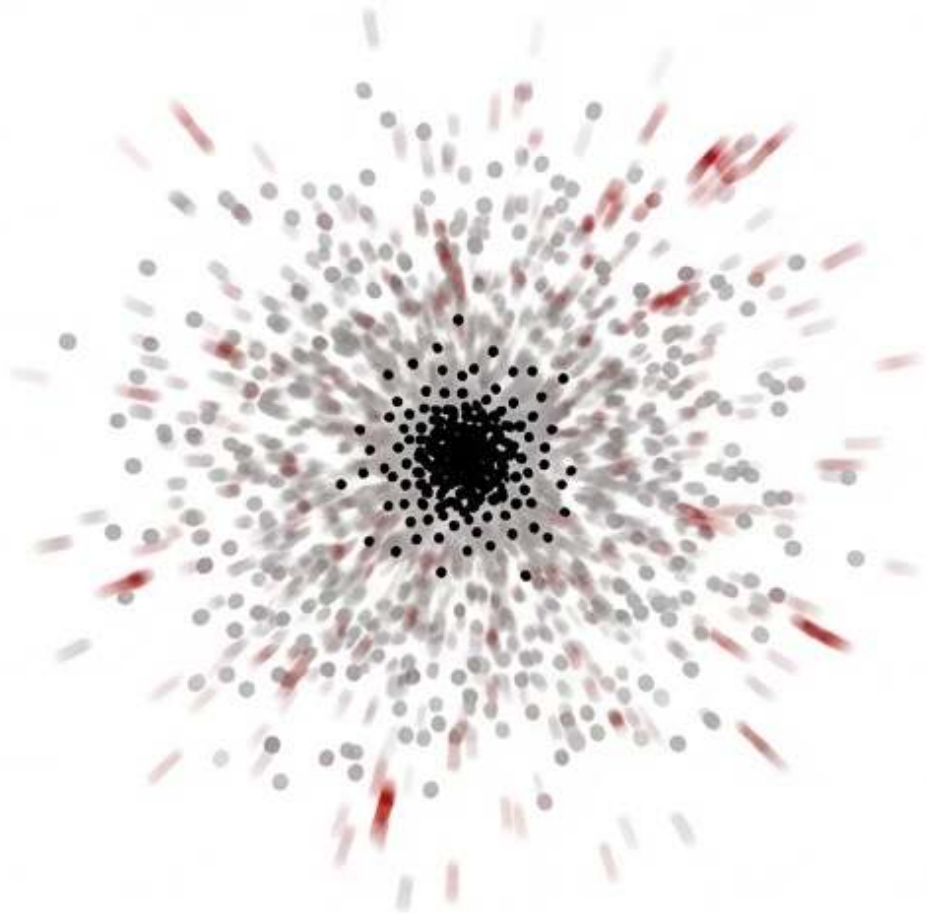
lines of code

Hugo Dutka Blog Post

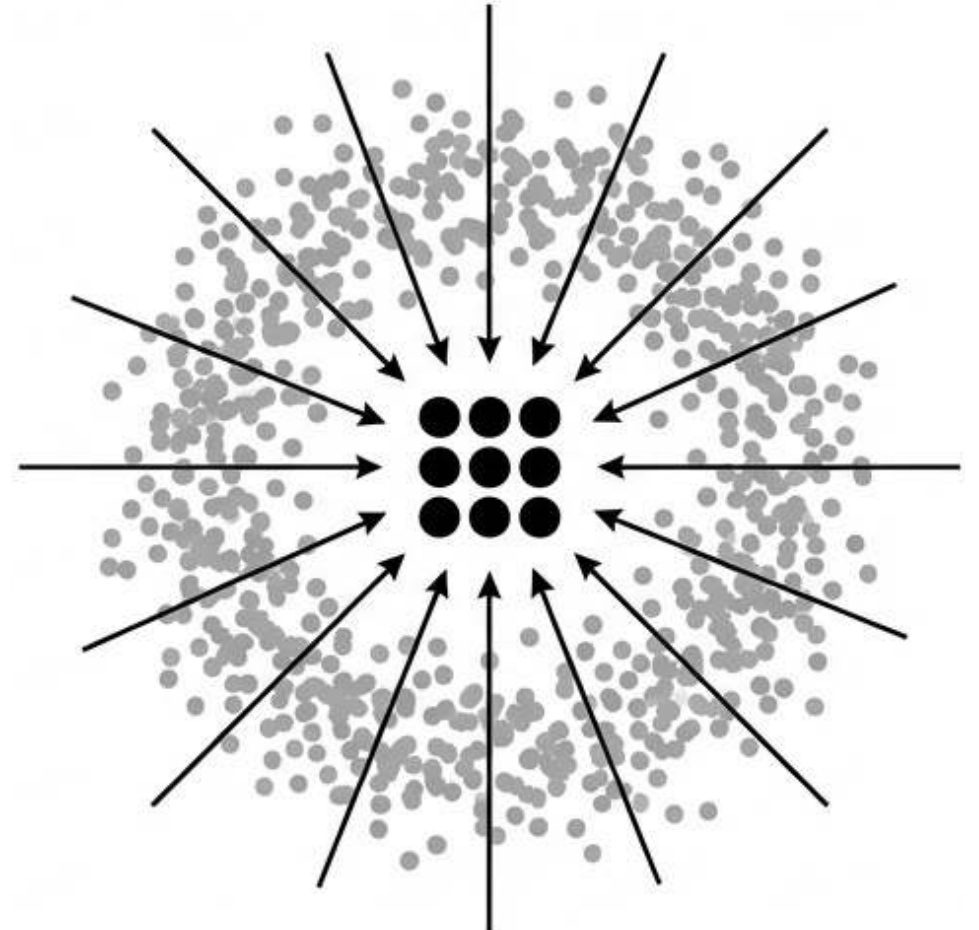


AI currently learns how to grow. It solves problems by adding code and layers.

Negentropic Capability



Entropy (Expansion)



Negentropy (Compression)

Negentropic Capability



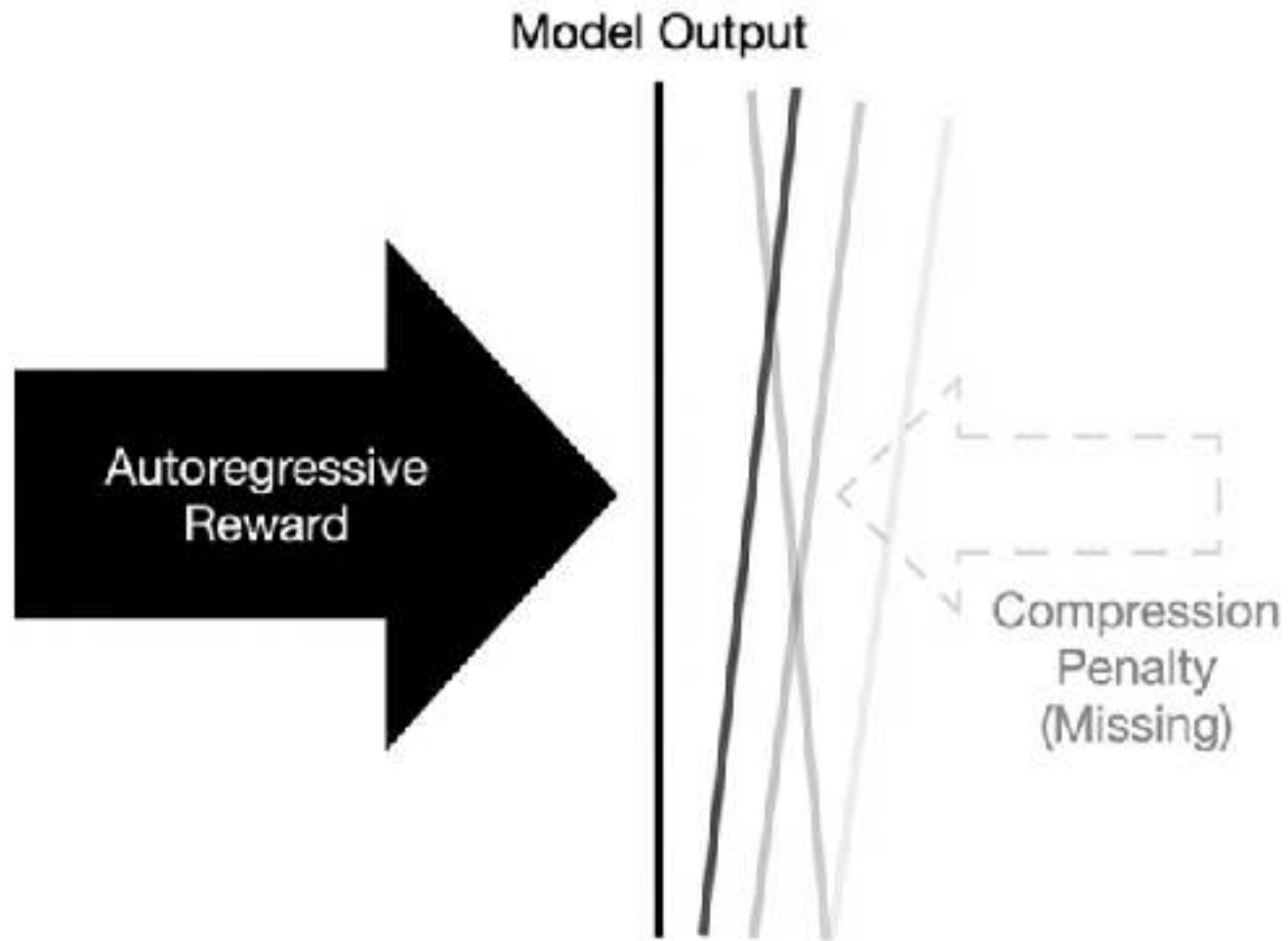
Schrödinger (1944): "Life feeds on negative entropy."

Living systems maintain order against entropy increase by extracting negentropy from the environment.

Autoregressive generation is an entropy engine.

It predicts the next token, perpetually expanding and unrolling.

Compression as Intelligence



What is intelligence?

On the Measure of Intelligence

François Chollet *

Google, Inc.

fchollet@google.com

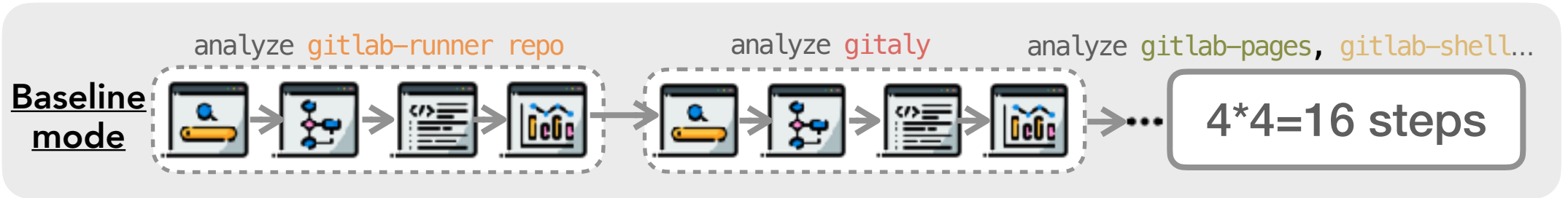
November 5, 2019

The intelligence of a system is a measure of its *skill-acquisition efficiency* over *a scope of* tasks, with respect to *priors, experience, and generalization difficulty*.

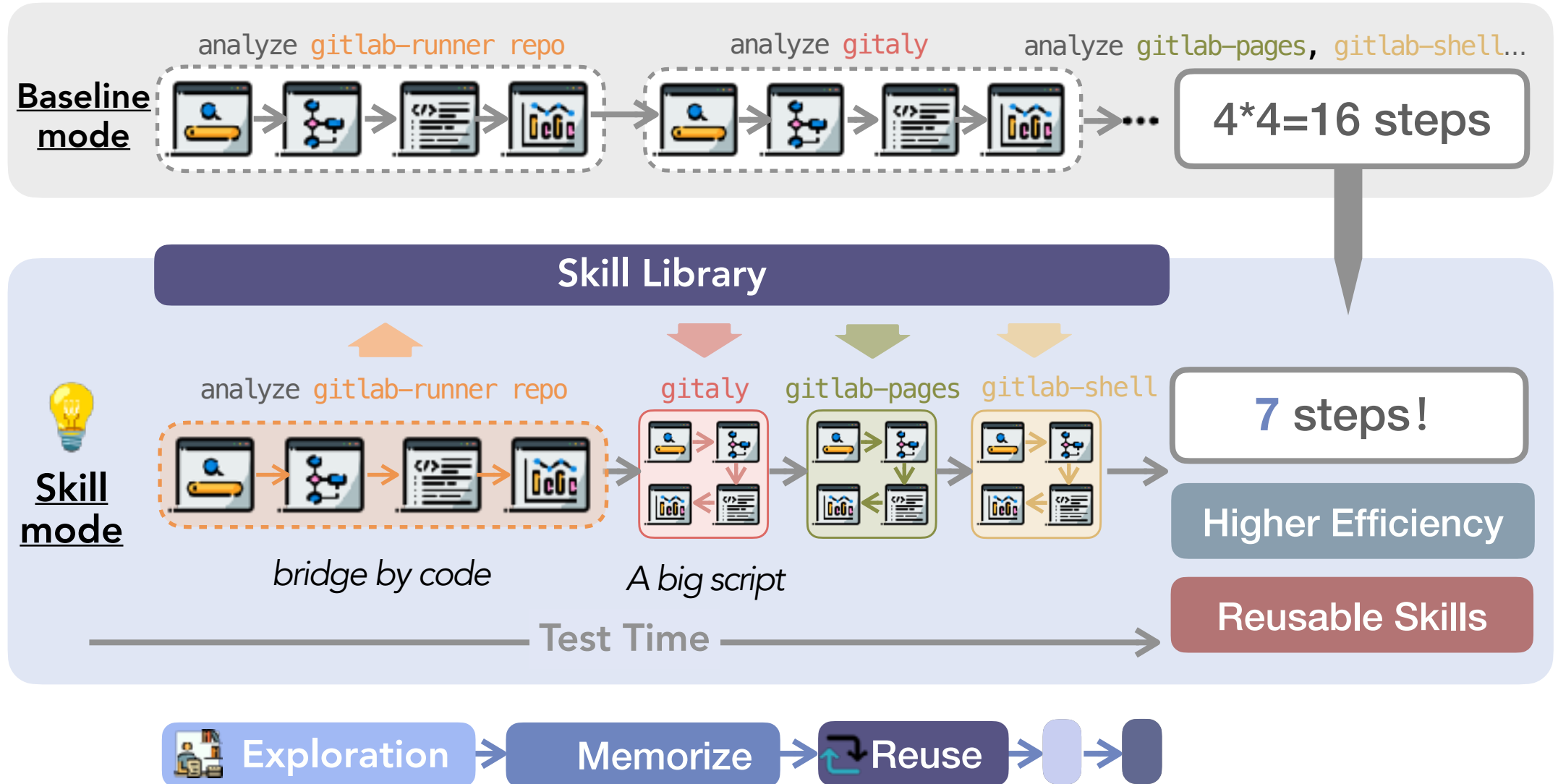
SkillCraft:

Can LLMs automatically **reuse skills**?

ON-THE-FLY Skill Composition



ON-THE-FLY Skill Composition



Skill Library



region_earthquakes_analyzer



vocabulary_card_builder



gitlab_project_analyzer



country_economic_analyzer



tvmaze_show_data_extractor



city_weather_analyzer



cocktail_menu_generator



pokemon_info_collector

Iterative Skill Composition

Skill

```
def gitlab_project_analyzer(path):  
    info = get_project_info(path)  
    contributors = get_contributors(path)  
    commits = get_commits(path)  
    branches = get_branches(path)  
    return {  
        "path": project_path,  
        "name": info["name"],  
        "stars": info["stars"],  
        "contributors":  
contributors["total"],  
        "recent_commits":  
commits["count"],  
        "branches": branches["total"],  
    }
```

Skill Library



region_earthquakes_analyzer



vocabulary_card_builder



gitlab_project_analyzer



country_economic_analyzer



tvmaze_show_data_extractor



city_weather_analyzer



cocktail_menu_generator



pokemon_info_collector

Test-Time Tool Chain Evolve

Atomic Tool

```
def get_project_info(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}")  
    return {"name": resp["name"], "stars": resp["star_count"],  
            "forks": resp["forks_count"]  
}
```

Code-wrapped Atomic Tool

```
def get_contributors(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/contributors")  
    return {"total": len(resp), "top_5": resp[:5]}  
}
```

```
def get_commits(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/commits")  
    return {"count": len(resp), "recent": resp[:20],  
            "authors": unique_authors(resp),  
}
```

```
def get_branches(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/branches")  
    protected = [b for b in resp if b["protected"]]  
    return {"total": len(resp),  
            "protected": len(protected),  
            "default": resp[0]["name"],  
}
```

Iterative Skill Composition

Skill

```
def gitlab_project_analyzer(path):  
    1 info = get_project_info(path)  
    2 contributors = get_contributors(path)  
    3 commits = get_commits(path)  
    4 branches = get_branches(path)  
  
    return {  
        "path": project_path,  
        "name": info["name"],  
        "stars": info["stars"],  
        "contributors":  
            contributors["total"],  
        "recent_commits":  
            commits["count"],  
        "branches": branches["total"],  
    }
```

Skill Library



region_earthquakes_analyzer



vocabulary_card_builder



gitlab_project_analyzer



country_economic_analyzer



tvmaze_show_data_extractor



city_weather_analyzer



cocktail_menu_generator



pokemon_info_collector

Test-Time Tool Chain Evolve

Atomic Tool

Code-wrapped Atomic Tool

```
def get_project_info(path):
    resp = gitlab_api(endpoint=f"/projects/{path}")
    return {"name": resp["name"], "stars": resp["star_count"],
            "forks": resp["forks_count"]}

def get_contributors(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/contributors")
    return {"total": len(resp), "top_5": resp[:5]}

def get_commits(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/commits")
    return {"count": len(resp), "recent": resp[:20],
            "authors": unique_authors(resp),}

def get_branches(path):
    resp = gitlab_api(endpoint=f"/projects/{path}/branches")
    protected = [b for b in resp if b["protected"]]
    return {"total": len(resp),
            "protected": len(protected),
            "default": resp[0]["name"],
            }
```



Tool Chain Execution

Task:
Analyze
multiple gitlab
repositories

Sample Task

Task Library

Iterative Skill Composition

Skill

```
def gitlab_project_analyzer(path):
    1 info = get_project_info(path)
    2 contributors = get_contributors(path)
    3 commits = get_commits(path)
    4 branches = get_branches(path)

    return {
        "path": project_path,
        "name": info["name"],
        "stars": info["stars"],
        "contributors":
            contributors["total"],
        "recent_commits":
            commits["count"],
        "branches": branches["total"],
    }
```

Skill Library

- region_earthquakes_analyzer
- vocabulary_card_builder
- gitlab_project_analyzer
- country_economic_analyzer
- tvmaze_show_data_extractor
- city_weather_analyzer
- cocktail_menu_generator
- pokemon_info_collector

Usable Skill retrieval

Test-Time Tool Chain Evolve

Atomic Tool

```
def get_project_info(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}")  
    return {"name": resp["name"], "stars": resp["star_count"],  
            "forks": resp["forks_count"]  
}
```

Code-wrapped Atomic Tool

```
def get_contributors(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/contributors")  
    return {"total": len(resp), "top_5": resp[:5]}  
}
```

```
def get_commits(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/commits")  
    return {"count": len(resp), "recent": resp[:20],  
            "authors": unique_authors(resp),  
}
```

```
def get_branches(path):  
    resp = gitlab_api(endpoint=f"/projects/{path}/branches")  
    protected = [b for b in resp if b["protected"]]  
    return {"total": len(resp),  
            "protected": len(protected),  
            "default": resp[0]["name"],  
}
```



Tool Chain Execution

Sample Task

Task Library

Task:
Analyze
multiple gitlab
repositories

Iterative Skill Composition

Skill

```
def gitlab_project_analyzer(path):  
    1 info = get_project_info(path)  
    2 contributors = get_contributors(path)  
    3 commits = get_commits(path)  
    4 branches = get_branches(path)  
  
    return {  
        "path": project_path,  
        "name": info["name"],  
        "stars": info["stars"],  
        "contributors":  
            contributors["total"],  
        "recent_commits":  
            commits["count"],  
        "branches": branches["total"],  
    }
```

Execute

Return

If Fail,
Re-explore



Multiple Env

Coding Verifier

If Pass,
Save new skill

Usable Skill retrieval

Skill Library



region_earthquakes_analyzer



vocabulary_card_builder



gitlab_project_analyzer



country_economic_analyzer



tvmaze_show_data_extractor



city_weather_analyzer

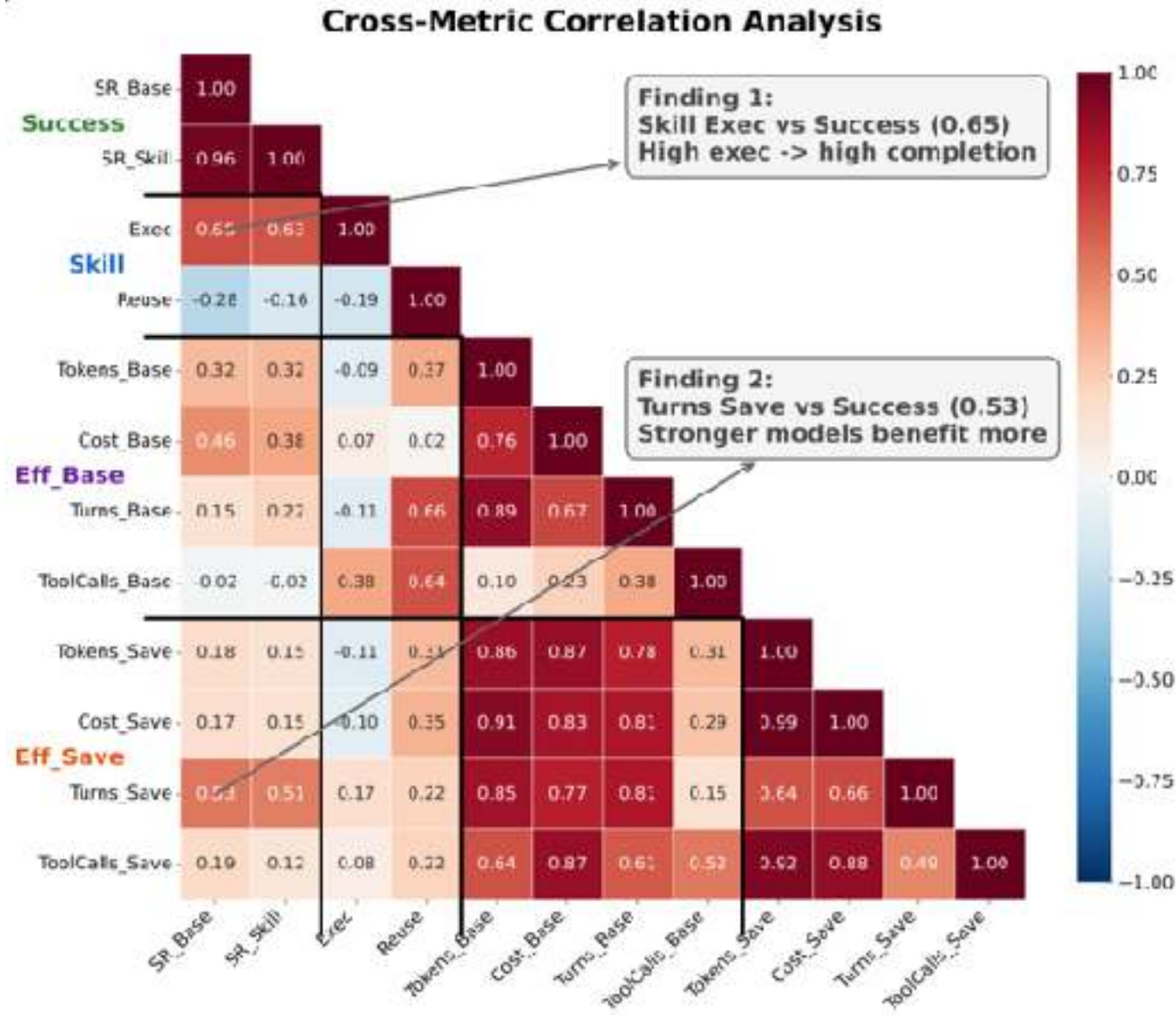


cocktail_menu_generator

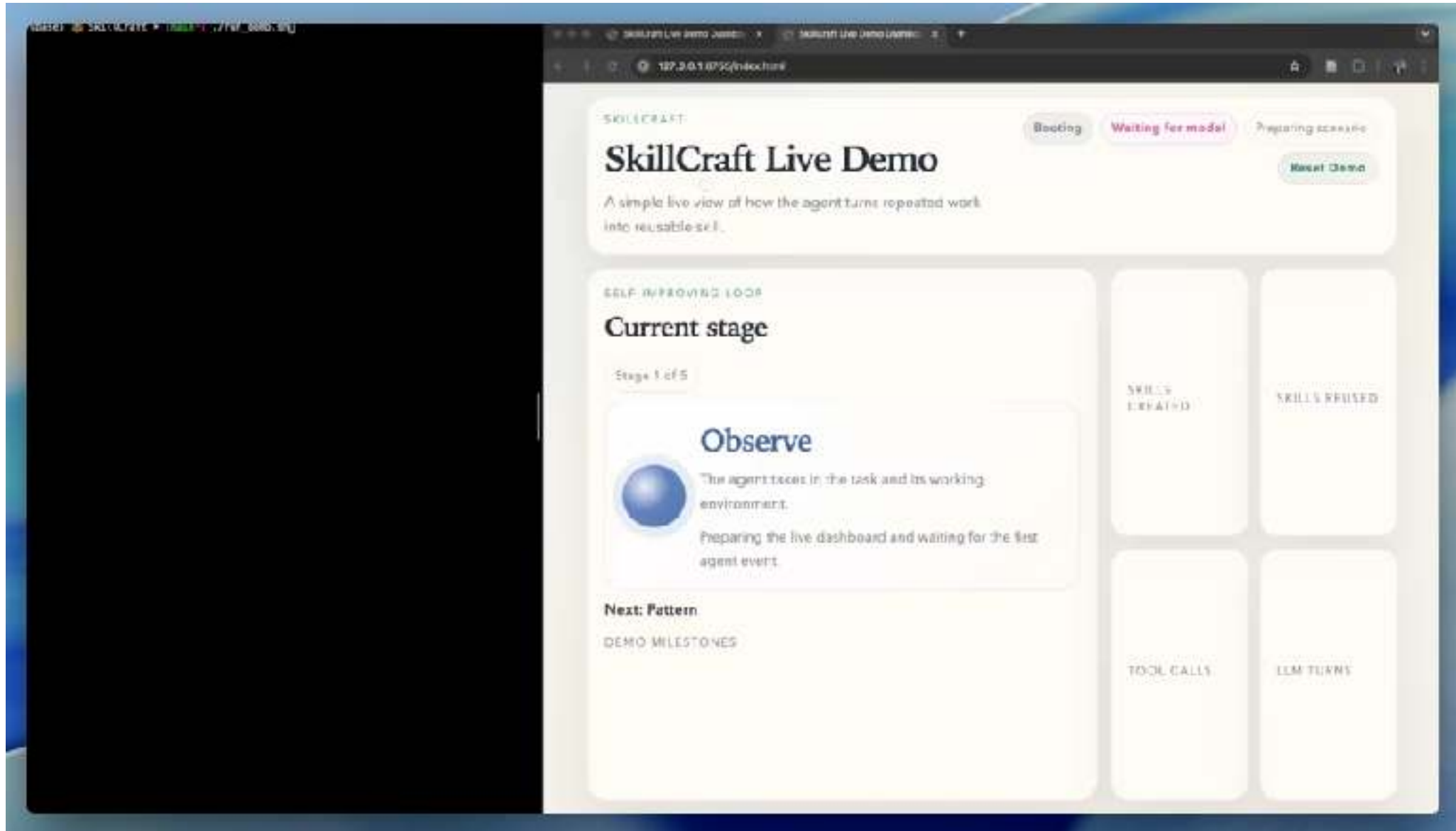


pokemon_info_collector

Can skills help?



Agentic Reasoning to discover Skills



Chen, ..., Li, He, Teh. SkillCraft (under review)

Budget-Aware Agents

Single-point estimation (Prior Work)



Question: Which college did the author of The Hobbit attend?



Cost: 1400 tokens



Progressive interval estimation (Ours)

Start

After T1

After T2

If unlikely

[700, 1200]

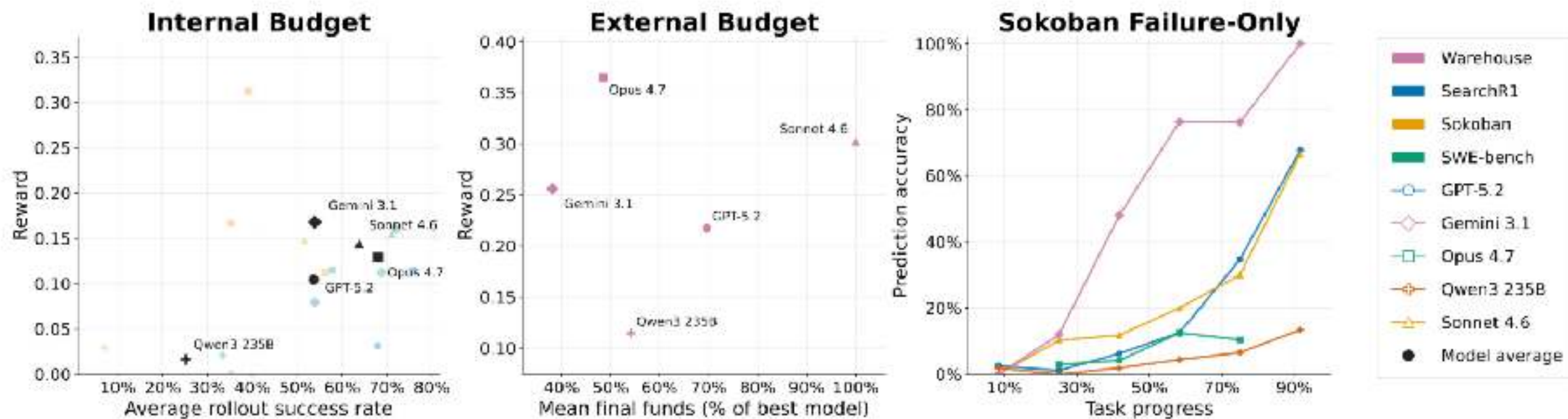
[450, 700]

[120, 250]

IMPOSSIBLE

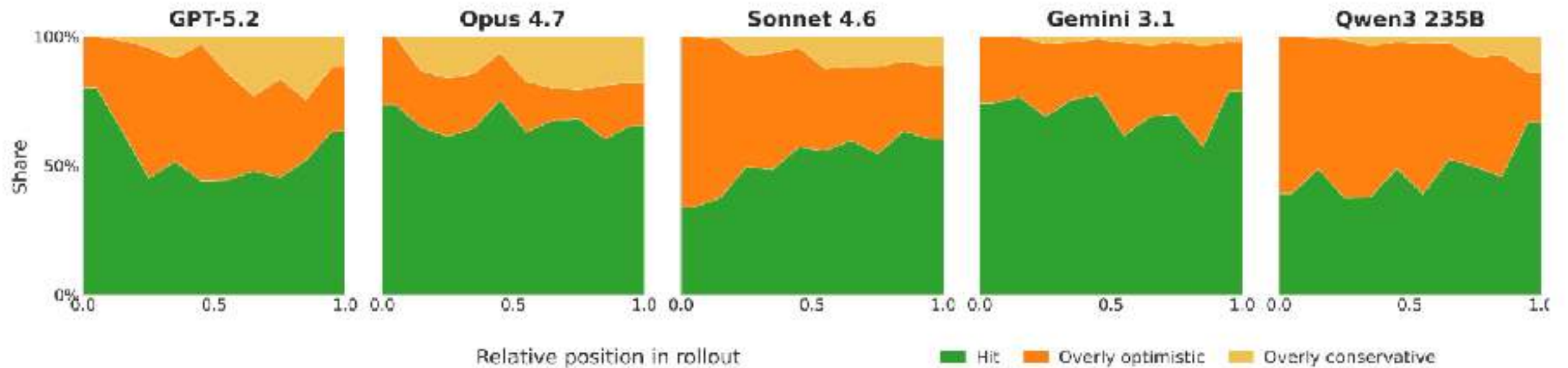
- ✓ Confidence-aware intervals
- ✓ Updated during interaction
- ✓ Output **IMPOSSIBLE** to warn early failure

Can Models Estimate Budget?

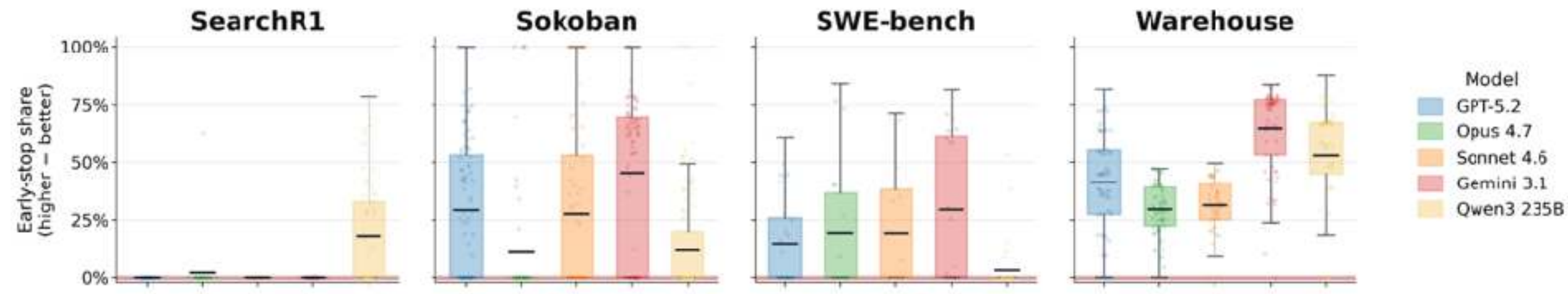


Task Success Rate - Budget Estimation: Correlation $r = 0.35$

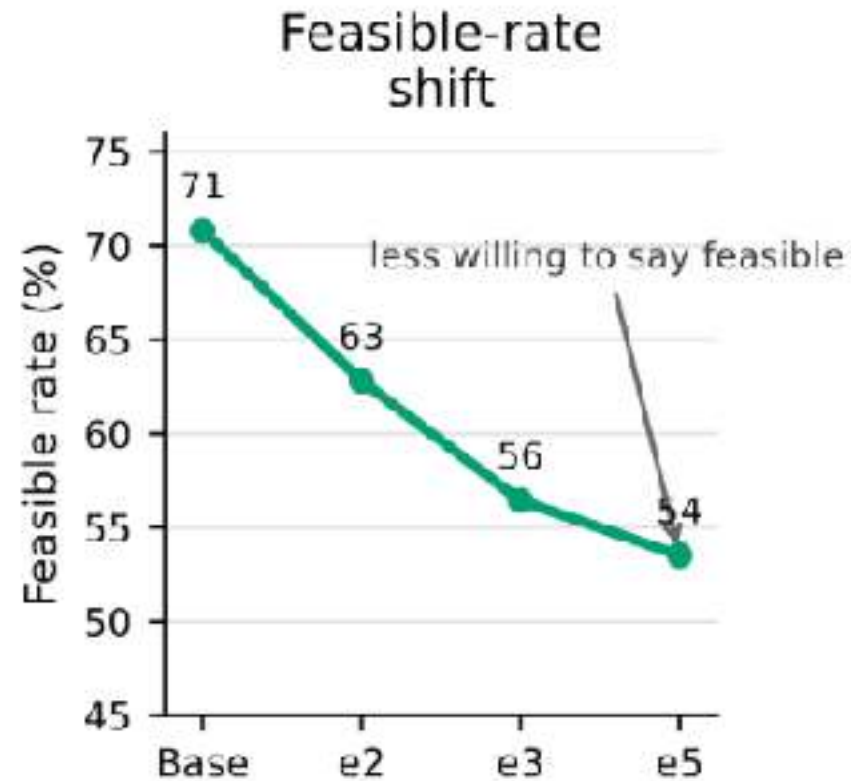
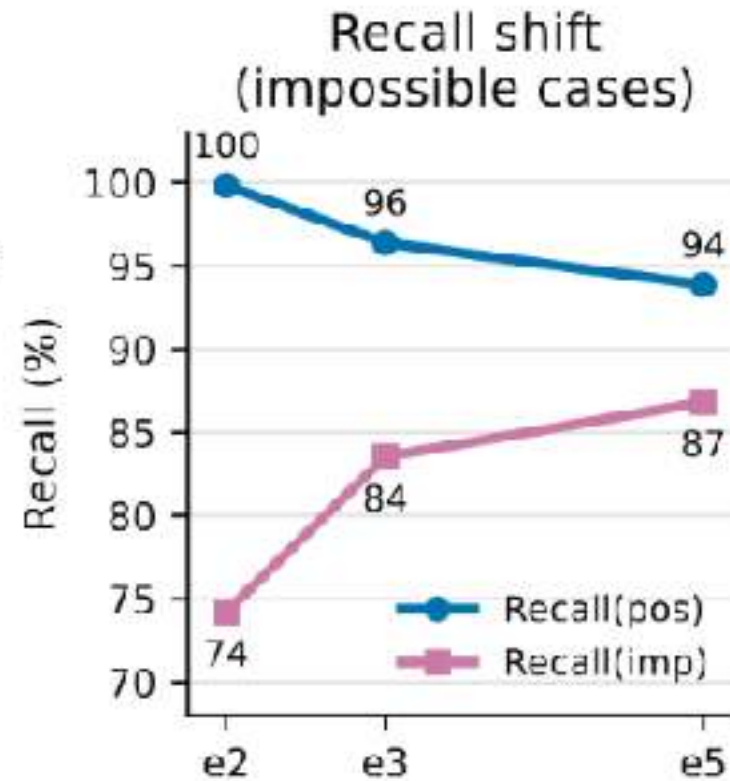
Overoptimistic: Failures are recognized late



Overoptimistic: Failures are recognized late



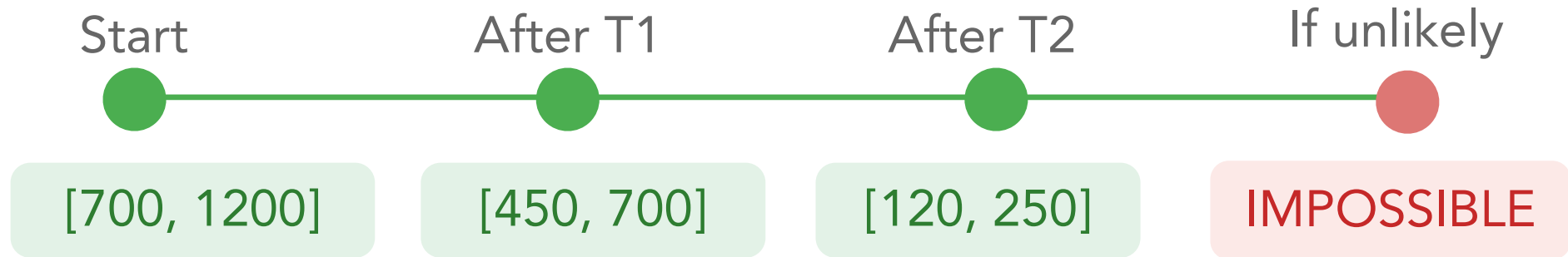
Simple Early-Stop SFT to teach model to stop early





Are Agents Budget-Aware?

Yuxiang Lin*, Zihan “Zenus” Wang*†, Mengyang Liu*, Yuxuan Shan*, Longju Bai*, Junyao Zhang, Xing Jin, Boshan Chen, Jinyan Su, Xingyao Wang, Jiaxin Pei, Manling Li



Progressive Interval Estimation

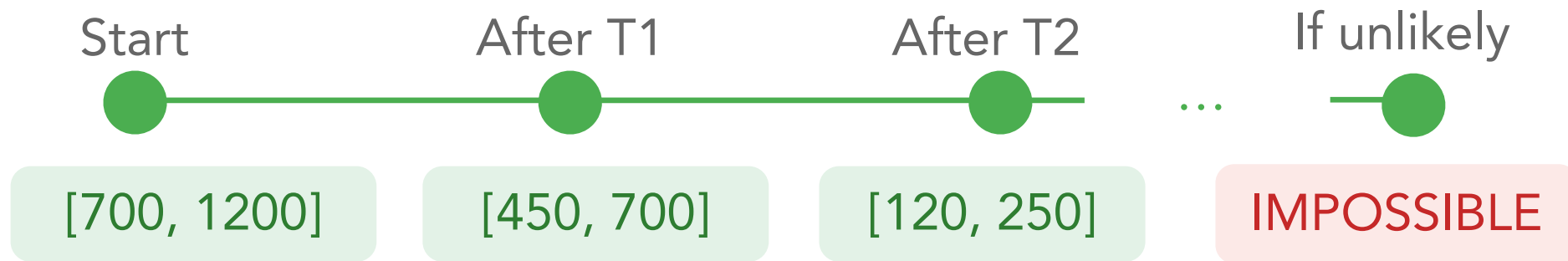
Over-Optimistic Failures

Actionable / Trainable to output **IMPOSSIBLE** to warn early failure



Are Agents Budget-Aware?

Yuxiang Lin*, Zihan “Zenus” Wang*†, Mengyang Liu*, Yuxuan Shan*, Longju Bai*, Junyao Zhang, Xing Jin, Boshan Chen, Jinyan Su, Xingyao Wang, Jiaxin Pei, Manling Li



Progressive Interval Estimation

Confidence-aware intervals

Over-optimistic

actionable / trainable to output **IMPOSSIBLE** to warn early failure

