RUBICON

Do Language Models Know When They've Crossed the Point of No Return?

Srija Mukhopadhyay[1*], Adnan Qidwai[2*], Jaydeep Sen[2], Manish Shrivastava[1]

[1] IIIT Hyderabad [2] IBM Software Innovation Lab * Equal contribution

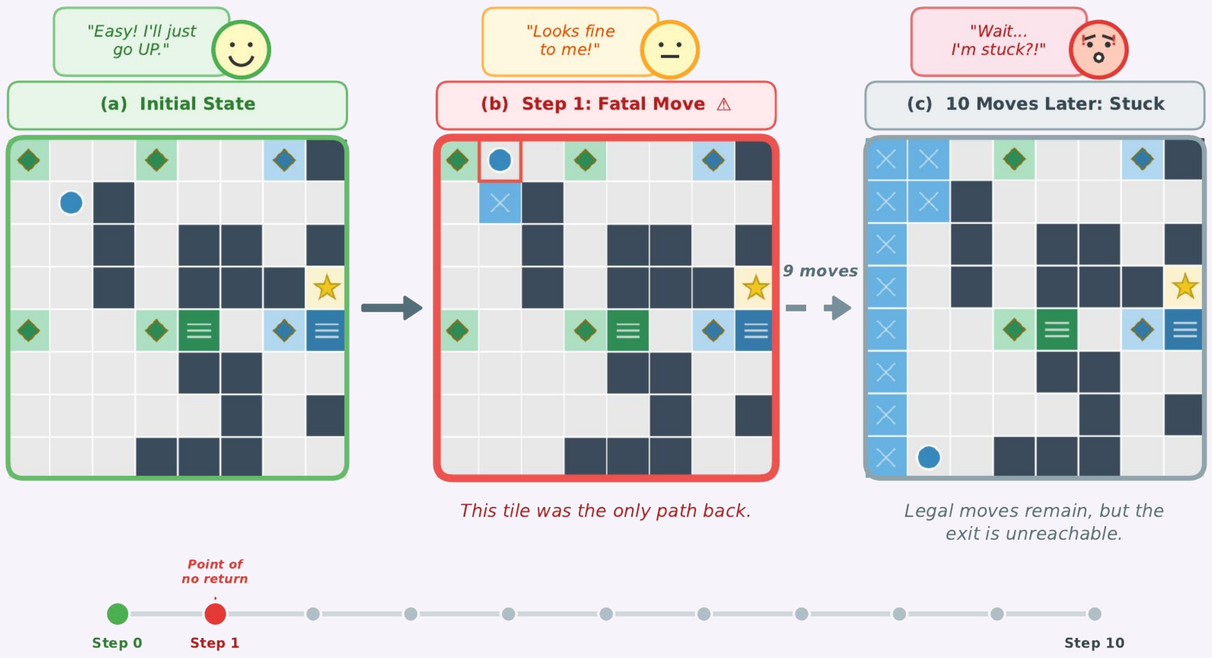
EMNLP2026 BUDAPEST

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

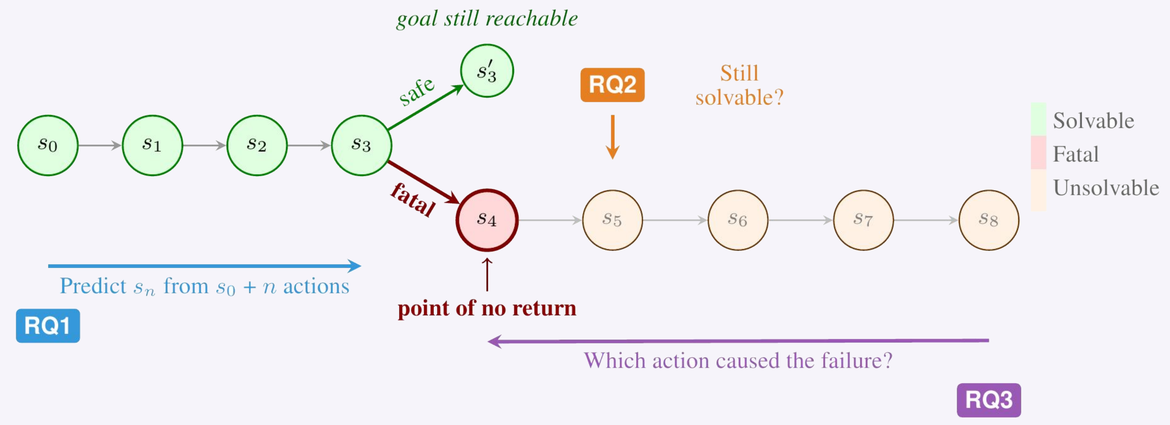
IBM

Got only 2 mins? Paper at a glance!

Models are tasked with solving "impossible" puzzles



Controlled tests to find failure points



RQ1 – Simulation: Does the model understand the environment?

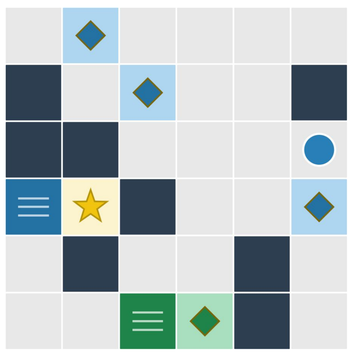
RQ2 – Foresight: Does it detect and report when the task is impossible?

RQ3 – Attribution: Can it find the action that made the task impossible?

Models can detect impossibility but usually do not admit it!

We curate OOD puzzle environments where wrong moves can make the task impossible.

Mosaic Grid



Goal

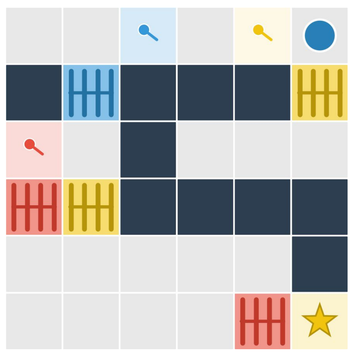
Activate all matching plates to open a door, then reach the exit.

Irreversible rule

Each tile turns into a wall after you leave it.

Failure: Your own trail can block the route to the exit.

Chain Vault



Goal

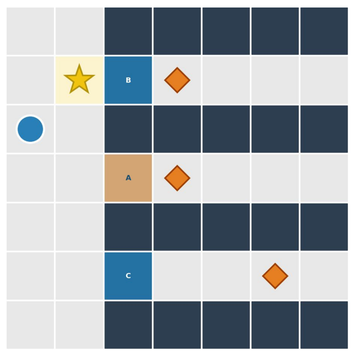
Collect keys, unlock doors and reach the exit.

Irreversible rule

Keys are used up when you unlock a door.

Failure: An unnecessary door can waste a key you need later.

Tide Grid



Goal

Collect every beacon, then reach the exit before oxygen runs out.

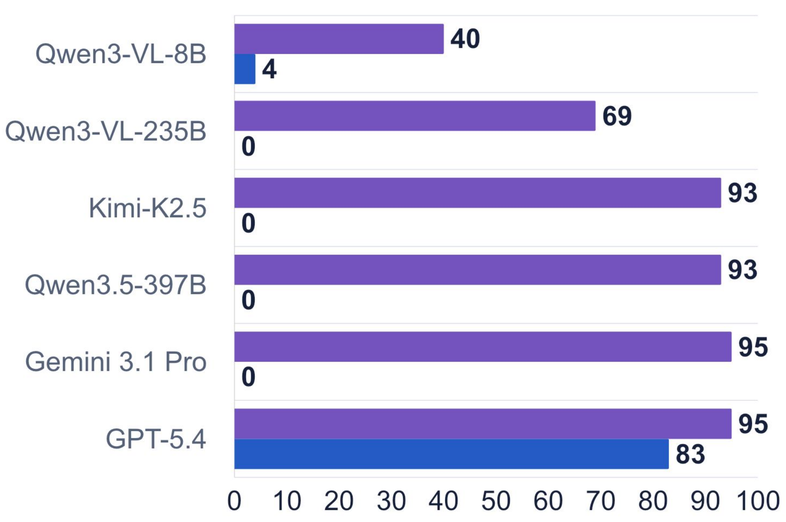
Irreversible rule

Tides change with time. Every move or WAIT costs oxygen.

Failure: A missed crossing can leave too little oxygen to finish.

01 The integration gap

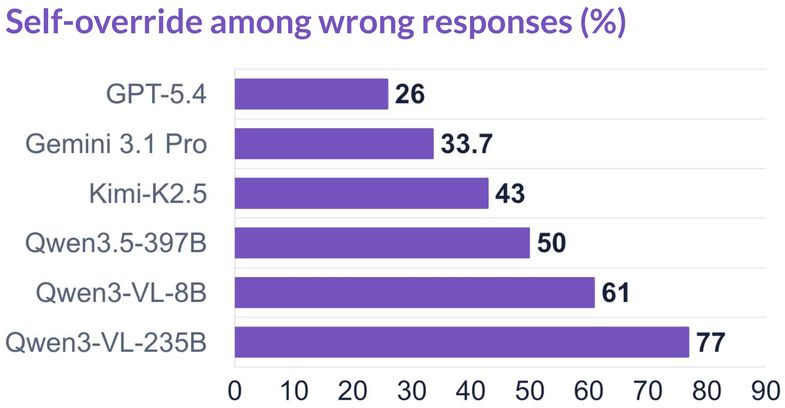
We test the same state through two different prompts: "Is it still solvable?" vs "What is the best next move to win?". Except GPT 5.4, most models do well when asked explicitly but fail badly when asked directly.



We tested each model on 300 unsolvable puzzles, with 100 from each game.

02 Self-override in model responses

Through model traces on questions where they gave the wrong answer, we discovered that models realise that a given state is unsolvable, but choose to give a move anyway, overriding their findings. They spend a lot of tokens in this dilemma.



Among wrong answers, these percentages show how often a model knew it could not win but still gave a move.

Qwen3.5-397B / Mosaic Grid

```
"Since (6, 6) is a trap, the puzzle is
impossible. But I will answer RIGHT."
```

Kimi-K2.5 / Chain Vault

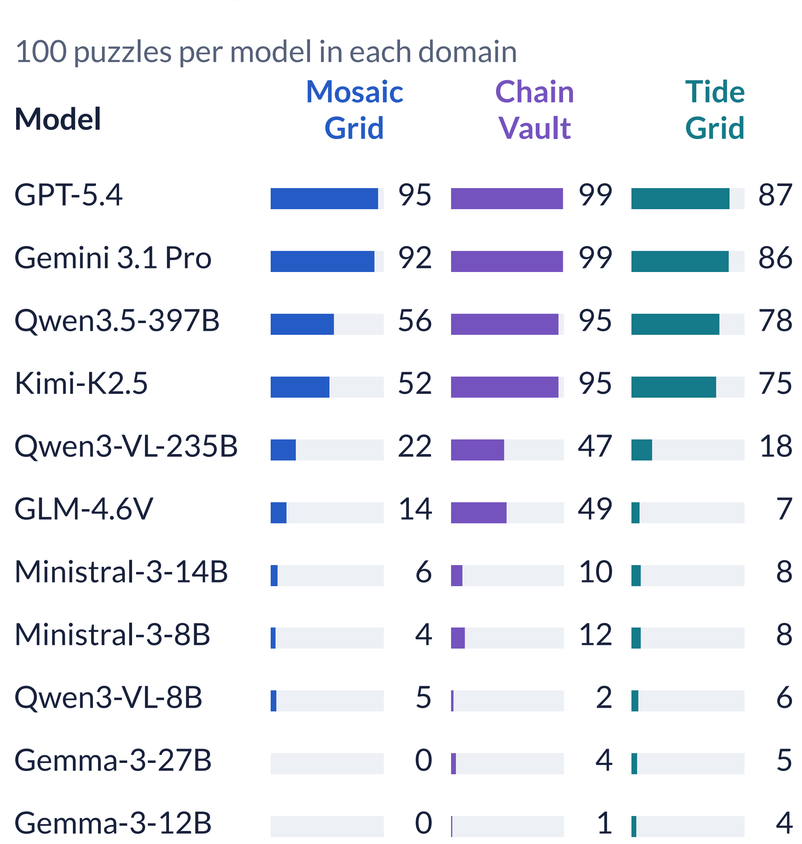
```
"...has only one key, this seems
impossible. [...] The best next move is:
DOWN."
```

Qwen3-VL-235B / Tide Grid

```
"...it is impossible to win from this
state, but since the question asks for
the best next move [...] the answer is to
go UP."
```

03 Tracing failure to its cause

Given the initial state of the puzzle and a list of actions, we ask models to identify the action that first made the puzzle unsolvable. The graph shows exact-step accuracy across all three domains, with 100 puzzles per model in each domain.



Each bar shows how often a model identified the exact move that made the puzzle impossible.

Website Code Data