MaxSAT-Based Feedback for Guiding Vision–Language Models in Sudoku

Pedro Orvalho[1] Guillem Alenyà[2] Felip Manyà[1]

[1]Artificial Intelligence Research Institute (IIIA–CSIC), Barcelona, Catalonia, Spain
[2]Institut de Robòtica i Informàtica Industrial (IRI-CSIC–UPC), Barcelona, Catalonia, Spain

EPIA 2026
Funchal, Madeira, September 2, 2026

IIIA-CSIC INSTITUT D'INVESTIGACIÓ EN INTEL·LIGÈNCIA ARTIFICIAL

CSIC Consejo Superior de Investigaciones Científicas

Marie Skłodowska-Curie Actions Postdoctoral Fellowship

Institut de Robòtica i Informàtica Industrial

The central question

Can formal constraint optimisation guide a Vision–Language Model (VLM) solving a visual logic puzzle?

The central question

Can formal constraint optimisation guide a Vision–Language Model (VLM) solving a visual logic puzzle?

What the VLM is good at

- Read the board image
- Propose plausible placements
- Produce natural-language explanations

The central question

Can formal constraint optimisation guide a Vision–Language Model (VLM) solving a visual logic puzzle?

What the VLM is good at

- Read the board image
- Propose plausible placements
- Produce natural-language explanations

What the VLM struggles with

- Maintain global consistency
- Recover from an early mistake
- Know which of many claims to retract

The central question

Can formal constraint optimisation guide a Vision–Language Model (VLM) solving a visual logic puzzle?

What the VLM is good at

- Read the board image
- Propose plausible placements
- Produce natural-language explanations

What the VLM struggles with

- Maintain global consistency
- Recover from an early mistake
- Know which of many claims to retract

Idea: Use a symbolic solver as a feedback oracle, not as a replacement brain.

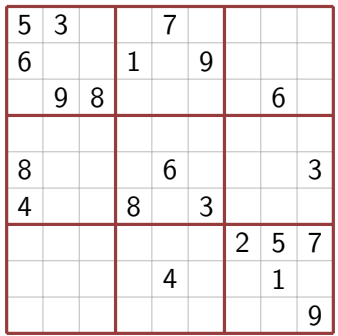
Why Sudoku is a useful testbed

- A visually structured task with a precise symbolic specification.

5	3			7				
6			1		9			
	9	8					6	
8				6				3
4			8		3			
						2	5	7
				4			1	
								9

Why Sudoku is a useful testbed

- A visually structured task with a precise symbolic specification.
- Every valid board satisfies cell, row, column, and 3×3 subgrid constraints.



Why Sudoku is a useful testbed

- A visually structured task with a precise symbolic specification.
- Every valid board satisfies cell, row, column, and 3×3 subgrid constraints.
- A symbolic solver can solve the logic problem quickly; the challenge here is the *neural proposal*.

5	3			7				
6			1		9			
	9	8					6	
8				6				3
4			8		3			
						2	5	7
				4			1	
								9

Why Sudoku is a useful testbed

- A visually structured task with a precise symbolic specification.
- Every valid board satisfies cell, row, column, and 3×3 subgrid constraints.
- A symbolic solver can solve the logic problem quickly; the challenge here is the *neural proposal*.
- This separates perception and reasoning failures in a controlled setting.

5	3			7				
6			1		9			
	9	8					6	
8				6				3
4			8		3			
						2	5	7
				4			1	
								9

Partial Maximum Satisfiability (MaxSAT)

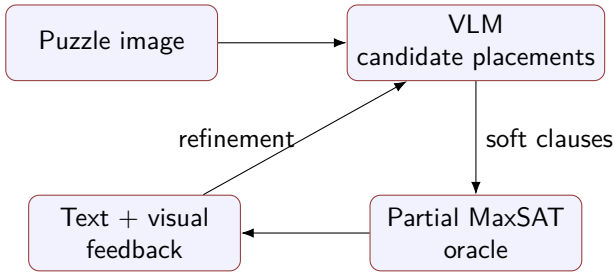
Hard:	$h_1 : (v_1 \vee v_2)$	$h_2 : (\neg v_2 \vee v_3)$
Soft:	$s_1 : (\neg v_1)$	$s_2 : (\neg v_3)$

Partial Maximum Satisfiability (MaxSAT)

Hard:	$h_1 : (v_1 \vee v_2)$	$h_2 : (\neg v_2 \vee v_3)$
Soft:	$s_1 : (\neg v_1)$	$s_2 : (\neg v_3)$
Possible Solution:	$\neg v_1 \ \vee \ v_2 \ \vee \ v_3$	$\text{cost} = 1$

Figure 1: Example of a partial MaxSAT formula.

Neuro-symbolic division of labour



- Neural component: perception and heuristic generation.
- Symbolic component: enforce constraints and identify a maximal consistent subset.

Sudoku as a partial MaxSAT problem

For each cell (r, c) and digit d, introduce $X_{r,c,d}$:

$$X_{r,c,d} = 1 \quad \Longleftrightarrow \quad \text{digit } d \text{ is placed at } (r, c).$$

Hard clauses

- Exactly one digit per cell
- Each digit once per row
- Each digit once per column
- Each digit once per subgrid

Soft clauses

- For each VLM proposal (r, c, d), add $(X_{r,c,d})$.
- MaxSAT satisfies all hard clauses while preserving as many proposals as possible.

What the oracle returns

Given proposals $\hat{A}$, solve

$$\hat{A}^* = \arg \max_{A \text{ satisfies Sudoku}} |A \cap \hat{A}|.$$

Interpretation

- $\hat{A}^*$: largest mutually consistent subset.
- $\hat{A} \setminus \hat{A}^*$: placements to revisit.

What the oracle returns

Given proposals $\hat{A}$, solve

$$\hat{A}^* = \arg \max_{A \text{ satisfies Sudoku}} |A \cap \hat{A}|.$$

Interpretation

- $\hat{A}^*$: largest mutually consistent subset.
- $\hat{A} \setminus \hat{A}^*$: placements to revisit.

Important distinction

Validity-only returns a textual **valid/invalid status**.

What the oracle returns

Given proposals $\hat{A}$, solve

$$\hat{A}^* = \arg \max_{A \text{ satisfies Sudoku}} |A \cap \hat{A}|.$$

Interpretation

- $\hat{A}^*$: largest mutually consistent subset.
- $\hat{A} \setminus \hat{A}^*$: placements to revisit.

Important distinction

Validity-only returns a textual **valid/invalid status**. MaxSAT additionally identifies the proposed placements that cannot belong to a largest consistent subset, with **textual and visual feedback**.

Two interaction protocols

Step-by-step

1. VLM proposes one placement.
2. Oracle checks it against the current state.
3. Accept or return corrective feedback.

Two interaction protocols

Step-by-step

1. VLM proposes one placement.
2. Oracle checks it against the current state.
3. Accept or return corrective feedback.

Full-board refinement

1. VLM proposes a complete board.
2. Oracle keeps the largest consistent subset.
3. Highlight conflicts and ask for a revised board.

Two interaction protocols

Step-by-step

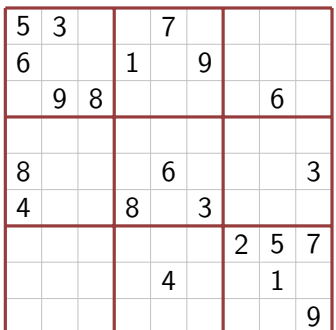
1. VLM proposes one placement.
2. Oracle checks it against the current state.
3. Accept or return corrective feedback.

Full-board refinement

1. VLM proposes a complete board.
2. Oracle keeps the largest consistent subset.
3. Highlight conflicts and ask for a revised board.

MaxSAT helps most when several proposals must be assessed together.

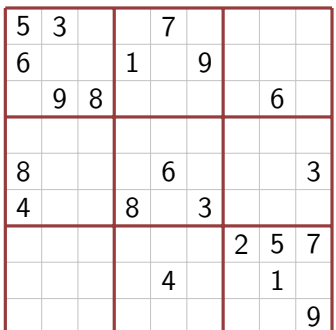
Running example: the initial puzzle



Running example

The VLM sees this image and proposes digits for blank cells.
The original clues are fixed;
Every new proposal becomes a soft clause.

Running example: the initial puzzle



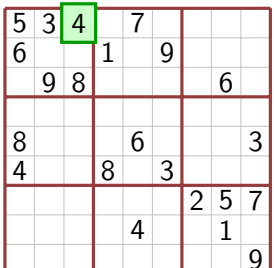
Running example

The VLM sees this image and proposes digits for blank cells.
The original clues are fixed;
Every new proposal becomes a soft clause.

- Valid next digit: $(r_1, c_3) = 4$.
- Invalid next digit: $(r_1, c_4) = 7$, which duplicates the clue at (r_1, c_5).

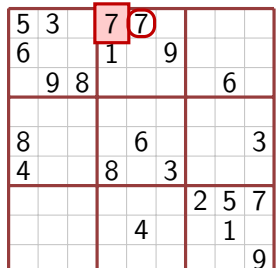
Running example: step-by-step feedback

Iteration 1: accept



Proposal $(r_1, c_3) = 4$ is valid.

Iteration 2: reject



Feedback: Reconsider (r_1, c_4).

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.
Four proposed digits are highlighted in red: they contradict the Sudoku constraints or other proposed digits.

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.
Four proposed digits are highlighted in red: they contradict the Sudoku constraints or other proposed digits.

- A validity-only oracle returns a textual invalid status

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.
Four proposed digits are highlighted in red: they contradict the Sudoku constraints or other proposed digits.

- A validity-only oracle returns a textual invalid status, but cannot say which digits must change.

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.
Four proposed digits are highlighted in red: they contradict the Sudoku constraints or other proposed digits.

- A validity-only oracle returns a textual invalid status, but cannot say which digits must change.
- Partial MaxSAT retains all compatible proposals

Running example: full-board proposal

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Full-board proposal

The VLM returns a complete board.
Four proposed digits are highlighted in red: they contradict the Sudoku constraints or other proposed digits.

- A validity-only oracle returns a textual invalid status, but cannot say which digits must change.
- Partial MaxSAT retains all compatible proposals, and identifies the minimum conflicting cells for the VLM to revisit.

The feedback regimes: text versus visual feedback

Validity-only

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Text feedback: "invalid board."

MaxSAT feedback

5	3	(7)	6	7	8	9	1	2
6	7	2	1	9	5	3	(5)	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	(9)	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
(6)	4	5	2	8	6	1	7	9

Text + visual: reconsider highlighted digits.

What changes for the VLM?

Validity-only: *text only* — *"the board is invalid"*.
MaxSAT: *text + visual* — *"revisit these highlighted placements"*.

Running example: full-board MaxSAT feedback

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Feedback returned to the VLM

- Keep the remaining 77 compatible placements.
- Reconsider the four highlighted cells.
- Produce a revised full-board proposal.

Running example: full-board MaxSAT feedback

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Feedback returned to the VLM

- Keep the remaining 77 compatible placements.
- Reconsider the four highlighted cells.
- Produce a revised full-board proposal.

Why is this useful?

The feedback **preserves reliable global structure**

Running example: full-board MaxSAT feedback

5	3	7	6	7	8	9	1	2
6	7	2	1	9	5	3	5	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	9	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
6	4	5	2	8	6	1	7	9

Feedback returned to the VLM

- Keep the remaining 77 compatible placements.
- Reconsider the four highlighted cells.
- Produce a revised full-board proposal.

Why is this useful?

The feedback **preserves reliable global structure** while focusing the next VLM call on the **smallest set of conflicting digits**.

Research questions

RQ1 To what extent **do VLMs produce placements that satisfy Sudoku** constraints without symbolic assistance?

Research questions

RQ1 To what extent **do VLMs produce placements that satisfy Sudoku** constraints without symbolic assistance?

RQ2 Does **MaxSAT-based feedback improve the ability of VLMs** to correct invalid moves and reach a complete, logically correct solution?

Research questions

RQ1 To what extent **do VLMs produce placements that satisfy Sudoku** constraints without symbolic assistance?

RQ2 Does **MaxSAT-based feedback improve the ability of VLMs** to correct invalid moves and reach a complete, logically correct solution?

RQ3 How do **open-source and closed-access VLMs differ in their ability to solve Sudoku** under iterative constraint-guided interaction?

Research questions

RQ1 To what extent **do VLMs produce placements that satisfy Sudoku** constraints without symbolic assistance?

RQ2 Does **MaxSAT-based feedback improve the ability of VLMs** to correct invalid moves and reach a complete, logically correct solution?

RQ3 How do **open-source and closed-access VLMs differ in their ability to solve Sudoku** under iterative constraint-guided interaction?

Evaluation focus

We compare validity-only and MaxSAT feedback for three VLMs in step-by-step and full-board interaction protocols.

Experimental design

- 200 Sudoku instances: 100 difficulty 0 and 100 difficulty 1; fixed seed 42.
- Three VLMs: GPT-5.5, Molmo (4B), and Qwen (32B).
- Two feedback regimes: validity-only vs. MaxSAT-based feedback.
- Temperature 0; 5 minutes per interaction and 30 minutes per puzzle.

Experimental design

- 200 Sudoku instances: 100 difficulty 0 and 100 difficulty 1; fixed seed 42.
- Three VLMs: GPT-5.5, MOLMO (4B), and QWEN (32B).
- Two feedback regimes: validity-only vs. MaxSAT-based feedback.
- Temperature 0; 5 minutes per interaction and 30 minutes per puzzle.
- Metrics:
  - # solved puzzles;
  - average completeness (% of cells matching ground truth).

Experimental design

- 200 Sudoku instances: 100 difficulty 0 and 100 difficulty 1; fixed seed 42.
- Three VLMs: GPT-5.5, MOLMO (4B), and QWEN (32B).
- Two feedback regimes: validity-only vs. MaxSAT-based feedback.
- Temperature 0; 5 minutes per interaction and 30 minutes per puzzle.
- Metrics:
  - # solved puzzles;
  - average completeness (% of cells matching ground truth).

Baseline intuition

The symbolic solver is not the bottleneck:

Experimental design

- 200 Sudoku instances: 100 difficulty 0 and 100 difficulty 1; fixed seed 42.
- Three VLMs: GPT-5.5, MOLMO (4B), and QWEN (32B).
- Two feedback regimes: validity-only vs. MaxSAT-based feedback.
- Temperature 0; 5 minutes per interaction and 30 minutes per puzzle.
- Metrics:
  - # solved puzzles;
  - average completeness (% of cells matching ground truth).

Baseline intuition

The symbolic solver is not the bottleneck: the MaxSAT oracle solves the encoded Sudoku instances in just a few seconds.

MaxSAT improves full-board refinement

Model	Validity only Solved	Validity only AC	MaxSAT feedback Solved	MaxSAT feedback AC
GPT-5.5	45	72.0%	**73**	**80.7%**
Molmo	17	45.4%	28	51.6%
Qwen	10	38.9%	13	42.6%

- GPT-5.5 solve rate rises from 22.5% to 36.5%.

MaxSAT improves full-board refinement

Model	Validity only Solved	Validity only AC	MaxSAT feedback Solved	MaxSAT feedback AC
GPT-5.5	45	72.0%	**73**	**80.7%**
Molmo	17	45.4%	28	51.6%
Qwen	10	38.9%	13	42.6%

- GPT-5.5 solve rate rises from 22.5% to 36.5%.
- Harder puzzles benefit strongly: GPT-5.5 solves 10 $\rightarrow$ 21 difficulty-1 instances.

MaxSAT helps in step-by-step mode too

Model	Validity only Solved	Validity only AC	MaxSAT feedback Solved	MaxSAT feedback AC
GPT-5.5	21	44.8%	**44**	**52.2%**
Molmo	5	35.9%	11	41.6%
Qwen	2	34.7%	6	36.5%

Observed Pattern

Every model improves in step-by-step mode too;

MaxSAT helps in step-by-step mode too

Model	Validity only Solved	Validity only AC	MaxSAT feedback Solved	MaxSAT feedback AC
GPT-5.5	21	44.8%	**44**	**52.2%**
Molmo	5	35.9%	11	41.6%
Qwen	2	34.7%	6	36.5%

Observed Pattern

Every model improves in step-by-step mode too;
The gain is largest when the initial proposal is already near-consistent.

Answers to RQ1–RQ3

- **RQ1 — without symbolic assistance:** VLM outputs are often incomplete or logically inconsistent, especially for the open-source models.

Answers to RQ1–RQ3

- **RQ1 — without symbolic assistance:** VLM outputs are often incomplete or logically inconsistent, especially for the open-source models.
- **RQ2 — effectiveness:** MaxSAT feedback improves both completeness and solve rate in both interaction protocols.

Answers to RQ1–RQ3

- **RQ1 — without symbolic assistance:** VLM outputs are often incomplete or logically inconsistent, especially for the open-source models.
- **RQ2 — effectiveness:** MaxSAT feedback improves both completeness and solve rate in both interaction protocols.
- **RQ3 — model effects:** all three VLMs benefit; GPT-5.5 remains strongest, while Molmo and Qwen also improve.

Answers to RQ1–RQ3

- **RQ1 — without symbolic assistance:** VLM outputs are often incomplete or logically inconsistent, especially for the open-source models.
- **RQ2 — effectiveness:** MaxSAT feedback improves both completeness and solve rate in both interaction protocols.
- **RQ3 — model effects:** all three VLMs benefit; GPT-5.5 remains strongest, while Molmo and Qwen also improve.

Key mechanism

Symbolic optimisation is **most useful when a coherent-but-imperfect global proposal contains a small set of conflicting placements**.

Where does the improvement come from?

1. **Preserve useful guesses:** correct placements are not discarded merely because another guess is wrong.

Where does the improvement come from?

1. **Preserve useful guesses:** correct placements are not discarded merely because another guess is wrong.
2. **Localise contradictions:** the solver exposes a small set of claims that cannot jointly hold.

Where does the improvement come from?

1. **Preserve useful guesses:** correct placements are not discarded merely because another guess is wrong.
2. **Localise contradictions:** the solver exposes a small set of claims that cannot jointly hold.
3. **Make revision actionable:** the VLM receives explicit cells to revisit rather than a generic invalidity message.

Where does the improvement come from?

1. **Preserve useful guesses:** correct placements are not discarded merely because another guess is wrong.
2. **Localise contradictions:** the solver exposes a small set of claims that cannot jointly hold.
3. **Make revision actionable:** the VLM receives explicit cells to revisit rather than a generic invalidity message.
4. **Repeat globally:** each refinement is checked against the complete Sudoku constraint system.

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements

Limits

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback

Limits

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface

Limits

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

- Quality depends on the initial proposal

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

- Quality depends on the initial proposal
- GPT/open-source performance gap remains

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

- Quality depends on the initial proposal
- GPT/open-source performance gap remains
- Sudoku is a puzzle with a **unique valid solution**.

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

- Quality depends on the initial proposal
- GPT/open-source performance gap remains
- Sudoku is a puzzle with a **unique valid solution**.
- Finding a solution for Sudoku is a **NP-complete problem**.

What the approach does—and does not—solve

Strengths

- Formal consistency guarantee for accepted placements
- Actionable, structured feedback
- Model-agnostic interface
- Strongest gains on full-board refinement

Limits

- Quality depends on the initial proposal
- GPT/open-source performance gap remains
- Sudoku is a puzzle with a **unique valid solution**.
- Finding a solution for Sudoku is a **NP-complete problem**.
- This explains the observed difficulty of VLMs in consistently producing complete and valid assignments.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.
- Encode domain rules as hard constraints.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.
- Encode domain rules as hard constraints.
- Treat proposals as soft constraints.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.
- Encode domain rules as hard constraints.
- Treat proposals as soft constraints.
- **Use MaxSAT to preserve the largest consistent subset and explain what must change**.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.
- Encode domain rules as hard constraints.
- Treat proposals as soft constraints.
- **Use MaxSAT to preserve the largest consistent subset and explain what must change**.

The complexity of solving logic puzzles motivates the **need for constraint-based refinement to support robust and explainable reasoning**.

Takeaway message

A VLM does not need to be a perfect solver to become more reliable.

- Let the VLM propose what it sees and hypothesises.
- Encode domain rules as hard constraints.
- Treat proposals as soft constraints.
- **Use MaxSAT to preserve the largest consistent subset and explain what must change**.

The complexity of solving logic puzzles motivates the **need for constraint-based refinement to support robust and explainable reasoning**.

Neural proposals + symbolic optimisation = more reliable visual reasoning

Thank you

Questions?

Pedro Orvalho, Guillem Alenyà, and Felip Manyà. **MaxSAT-Based Feedback for Guiding Vision–Language Models in Sudoku**. EPIA 2026.

This project has received funding from the European Union's HORIZON-MSCA-2025-PF research and innovation programme under the Marie Skłodowska-Curie (Sherlock4Py, GA No 101269051). This work was supported by grants PID2022-139835NB-C21 and PID2025-171340NB-I00 funded by MCIN/AEI/10.13039/501100011033. This project was additionally supported by the ALLIES Cofund, and has received funding from the European Union's Horizon-MSCA-2022-COFUND-01 research and innovation programme under the Marie Skłodowska-Curie (GA No 101126626). G. Alenyà acknowledges the support of EU project FlexCycle HORIZON-CL4-2024-DIGITAL-EMERGING-01-101189600.