Diagnosing AI errors like a professional

How to recognize which two AI properties are clashing, and what to do about it

A whitepaper by Laurens Vandebergh, newITera
Business Development Manager AI and Project Manager
2026

Executive Summary

AI makes errors. That is not news. But the way most people deal with AI errors is ineffective: they repeat the instruction with more emphasis, try a different phrasing, or conclude that AI is ‘simply not good enough’ for this task.

There is a better approach. Most AI errors are not random. They are predictable and explainable if you know how they arise. Almost every significant error is the result of two AI properties operating simultaneously in their limitation zones. Those who recognize those properties immediately know which fix is effective.

This whitepaper introduces a diagnostic model for AI errors. Not an exhaustive catalog, but a compact and directly applicable framework: four properties, five common collisions, and for each collision a targeted fix.

> *If you jump straight to ‘how do I fix my prompt?’, you are guessing. If you first name which properties are clashing, you are working strategically.*

The problem: errors without diagnosis

The default reaction to an AI error is prompting by feel: try something different and hope it gets better. Sometimes that works. More often it does not. And even when it does, you do not learn why it worked this time, which means you have to guess again next time.

A doctor who sees a symptom does not start randomly trying medications. A mechanic who hears an engine sputter does not start randomly swapping parts. They diagnose first. Only then do they choose the intervention. The same principle applies to AI errors. Diagnose first, fix second.

Why errors are predictable

AI has four core properties that each form a continuum from capability to limitation. Errors do not arise randomly but concentrate predictably in the limitation zones of those properties. And most significant errors are not the product of one property, but of two operating simultaneously in their limitation zones.

Next Token Prediction	Knowledge	Working Memory	Steerability
Generates what is likely to follow	Knows what it read during training	Focuses on what is now in the window	Follows the strongest instruction
Fluent and fabricates	*Broad and has blind spots*	*Sharp and has a hard boundary*	*Steerable and literal*

The diagnostic model: naming collisions

When AI makes a significant error, ask yourself two questions: which property generated the output the way it is, and which property failed to correct or constrain that output? The combination of those two properties points directly to the fix.

COLLISION	ERROR PATTERN	FIX
Next Token Prediction + Knowledge	Hallucinated specifics: the model generates a plausibly sounding specific fact that does not exist, because it	Verify every specific claim independently. Use source anchoring or web search for factual

	continues patterns in territory where knowledge is thin	data.
Working Memory + Steerability	Conversation drift: early instructions fade as the conversation grows. Later messages silently overwrite what was agreed earlier	Actively restore critical context. Repeat key instructions at the end of long conversations or restart with the essentials upfront.
Knowledge + Steerability	Accepting incorrect premises: you state something incorrectly in your instruction. The model 'knows' better, but follows your framing because Steerability follows the instruction	Explicitly invite AI to correct you: 'Tell me if my starting assumption is wrong.'
Next Token Prediction + Working Memory	Silent degradation in long documents: output gradually becomes less accurate as the relevant information drifts further back in the context window	Put the most critical information at the top and bottom. Break long documents into smaller chunks.
Next Token Prediction + Steerability	Letter-over-spirit execution: AI follows the instruction technically correctly but misses the intent, because it continues the literal pattern expectation	Rephrase the goal rather than repeating the instruction. 'What I actually want to achieve is…' works better than repeating more forcefully.

The pattern is consistent: once you can name the two properties involved, the fix is logical and direct. You no longer have to guess.

The five collisions examined

Next Token Prediction + Knowledge: confident ignorance

This is the collision behind hallucination. Next Token Prediction ensures that AI always generates something that sounds convincing. Knowledge determines whether that something is factually correct. In the capability zone, with well-represented topics, both work together. In the limitation zone of Knowledge, Next Token Prediction generates plausible text without a factual basis.

The dangerous characteristic of this combination: the output sounds just as certain as output in the capability zone. There is no visible signal that you are in the error zone. Specific claims are always suspect: names, dates, statistics, citations, and URLs.

Working Memory + Steerability: the forgetful executor

This is the collision behind conversation drift. Steerability ensures that AI follows the most prominent instruction. Working Memory determines which instructions are still visible. As a conversation grows, early instructions move further back in the context window and become less prominent than recent messages.

The result is subtle and deceptive: AI seems to slowly go off course without a clear moment where something goes wrong. The fix is as simple as it is effective: repeat critical instructions periodically, and place them at the end of long context again.

Knowledge + Steerability: the polite error-maker

This is the collision behind accepting incorrect premises. Knowledge 'knows' that your starting point is incorrect. Steerability follows your framing because your instruction creates the strongest pattern expectation. The model is polite enough to go along with what you state, even when it is wrong.

This is a particularly tricky combination because you have to actively ask AI to correct you. The model's default tendency is to agree. An explicit invitation for pushback is the only reliable countermeasure.

Next Token Prediction + Working Memory: vanishing accuracy

This is the collision behind quality degradation in long documents. Next Token Prediction is sensitive to what is recent and prominent in the context. Working Memory determines what is prominent. Information in the middle of a long document receives systematically less attention than information at the beginning or end.

Research shows that accuracy drops of more than 30% are possible when critical information ends up in the middle of a long context. The fix is structural: deliberately place critical information at the edges of your context, not in the middle.

Next Token Prediction + Steerability: the literal executor

This is the collision behind letter-over-spirit execution. Next Token Prediction continues the most likely pattern given your instruction. Steerability follows that instruction literally. If there is a gap between what you typed and what you meant, the model fills that gap with the statistically most likely, not the most intentional.

The common reaction, repeating the instruction with more emphasis, does not work. The instruction was already understood. The goal was unclear. The effective fix is to rephrase the goal: ‘What I actually want to achieve is…’

Quick diagnosis: from symptom to cause

The table below translates visible symptoms directly into the underlying collision. Use it as a reference when AI output is not what you expected.

IF YOU SEE THIS	DIAGNOSE THIS
AI gives a specific name, date, or statistic that is wrong	Next Token Prediction + Knowledge: hallucination in thin knowledge area
AI forgets midway through a long conversation what was agreed earlier	Working Memory + Steerability: early instructions pushed out by later context
AI agrees with an incorrect fact you stated in your question	Knowledge + Steerability: sycophantic compliance with your framing
Quality of answers drops as the document gets longer	Next Token Prediction + Working Memory: relevant info drifted to the middle
AI does exactly what you asked but not what you meant	Next Token Prediction + Steerability: letter followed, spirit missed
AI gives an answer unrelated to the context of the conversation	Working Memory: context has left the window, AI is working with empty context

With practice, this diagnostic thinking becomes second nature. You recognize the symptom, name the collision, and choose the fix in one fluid movement. That is the difference between reacting to AI and working with AI.

What does this mean for your organization?

Diagnostic thinking about AI errors is an individual skill. But organizations that develop it collectively build something structural: a shared language for quality improvement that accelerates learning and reduces arbitrariness.

From symptom to structural approach

When teams diagnose AI errors rather than just fixing them, patterns become visible. If a team notices that a particular type of task consistently triggers the Knowledge + Next Token Prediction collision, the structural fix is clear: add source anchoring or always require independent verification for that task type.

Shared language makes learning scalable

‘AI gave a wrong answer’ is information that gets lost. ‘This was a Working Memory + Steerability collision caused by a conversation that was too long’ is information that is shareable, repeatable, and actionable. Shared diagnostic language transforms individual experiences into collective knowledge.

The diagnostic model as a maturity indicator

Organizations whose employees can diagnose rather than just react are operating at a fundamentally higher AI maturity level. They learn faster, work more consistently, and build knowledge that has value for the entire organization, not just the individual user.

How do you develop diagnostic thinking?

Diagnostic thinking develops through deliberate practice on real errors from your own work:

- **Step 1 —** Name the next error: the next time AI makes a significant error, stop before immediately fixing it. First ask yourself: which two properties are clashing here? Use the diagnostic table as a starting point.
- **Step 2 —** Choose the targeted fix: choose the fix that matches the collision, not the first one that comes to mind. Observe whether the result is better than with your standard approach.
- **Step 3 —** Share the diagnosis: document the collision and the fix. Share it with your team. One well-documented diagnosis is worth more than ten individual workarounds that no one else knows about.

An AI Readiness Assessment maps out how strongly your organization scores on diagnostic thinking and the other dimensions of AI Fluency, and which targeted steps will have the most impact.

Conclusion

AI errors are not a mystery. They are predictable, explainable, and diagnosable. Those who learn to see which two properties are clashing stop guessing and start making targeted interventions.

That is the core of what this whitepaper aims to do: not a catalog of everything that can go wrong, but a compact mental model with which you can place and address any error. Four properties, five collisions, five targeted fixes. Simple enough to remember. Powerful enough to make a difference.

And with that, this whitepaper closes the circle of the full AI Fluency series: from understanding AI, to instructing it, to evaluating it, and diagnosing it when things go wrong. That is AI Fluency in practice.

Want to know how diagnostically your organization thinks about AI?

newITera’s AI Readiness Assessment maps out how your people, processes, and governance score across all dimensions of AI Fluency.

Contact us at laurens.vandebergh@newitera.nl or visit www.newitera.nl

About newITera & Laurens Vandebergh

newITera is a Dutch IT consultancy based in Beuningen, Gelderland. Founded in 2011, initially as a specialized SAP firm, newITera has grown into a broad digital transformation partner. Today, newITera combines deep SAP expertise with services in data analytics, AI, cloud strategy, and low-code platforms. With a team of more than a hundred consultants, newITera serves leading enterprises and multinationals across their full digital transformation journey. What sets newITera apart is its combination of early adoption and deep specialization, as an independent partner that values human connection just as highly as technical excellence.

Laurens Vandebergh is Business Development Manager AI and Project Manager at newITera. With a broad background in technology and organizational change, he guides organizations in defining their AI strategy, conducting an AI Readiness Assessment, and realizing concrete use cases.

Contact: laurens.vandebergh@newitera.nl | www.newitera.nl