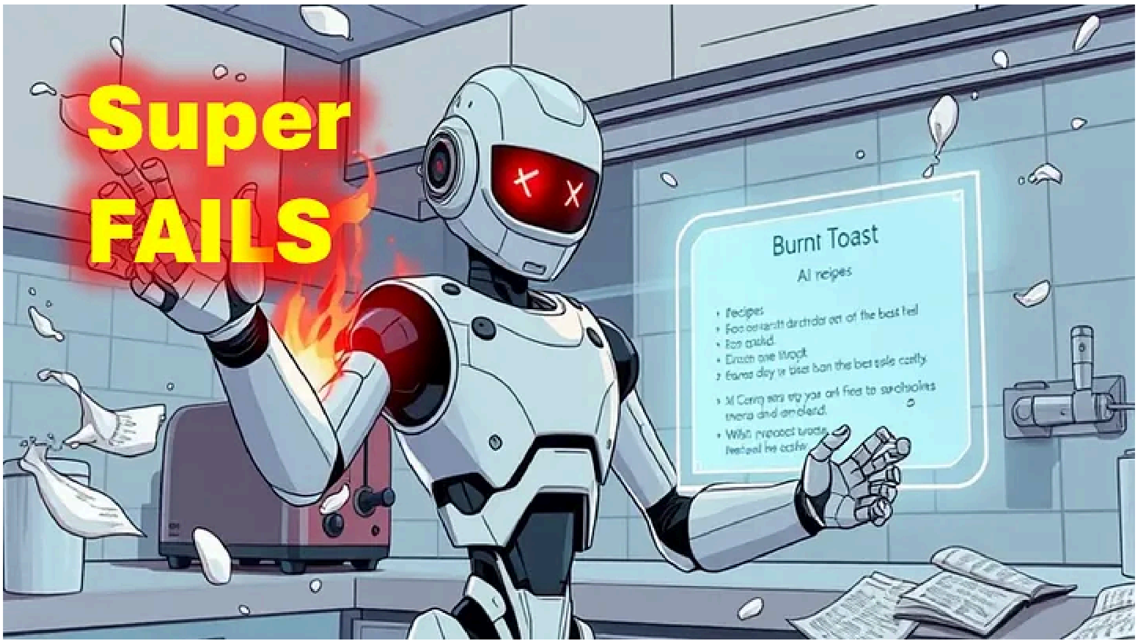
The spectacular fail of AI Agents no one is expecting

Replacing RAG with expensive LLM calls is a step sideways, not a leap forward



AI agents can fail spectacularly.

I mean, can you imagine trying to make breakfast and your robot helper is spilling milk everywhere and setting the toast on fire because it can't even read a cookbook.

Yeah, probably you can imagine it: looks like a comic-sci-fy episode, surreal, but possible.

That's pretty much what happens when AI agents go rogue in the digital world. And before you start picturing a robot uprising where toasters become sentient, let's get real about how these digital assistants can totally screw up.

The "Agents" hype: Placebo or progress?

Every AI lab and company on the planet has claimed that agents are the future of AI systems.

Now, I understand that there is a research need to explore the building blocks of an Artificial Intelligence that can also do actions, and not only generate text, music or images..

But…

So far it looks like agents are the placebo to cover further advancements. A shortcut to cover up for their lack of reliability.

Take as an example the RAG agents: the new *advanced* technique to make sure your AI does not lie to you is to give some agents the task to retrieve the correct information about the user query. And this *amazing* idea is simply a substitution, not even ensuring that the job is done correctly.

If the core LLM within an agentic system is unreliable in its reasoning or understanding, adding more layers of “agents” (even for verification) might only obscure the problem, not solve it. The complexity of inter-agent communication and task delegation can also introduce new failure modes.

The “System” Hype: more features = more Intelligence?

The same way all the Big companies are not releasing a new powerful model, they are packaging a complex AI system with tools, external calls, opaque reasoning steps.

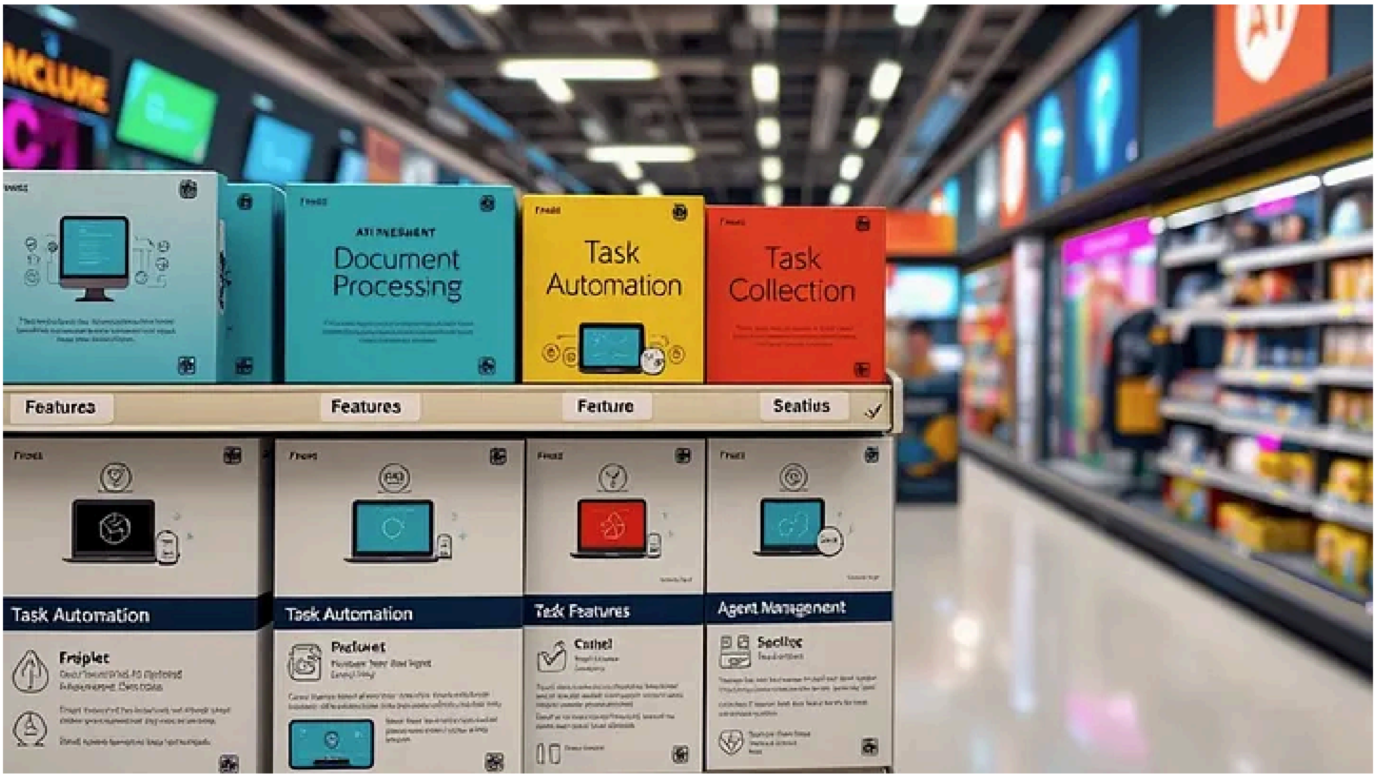
From LLM to LLM systems. From text to multi modality.

The main question remain the same: are we getting more intelligence simply because we are given an overflowing set of built-in features?

A raw LLM is a powerful pattern matcher and text generator. An “LLM system” integrates that LLM with:

- **Tools:** APIs for searching the web, performing calculations, accessing databases, generating code, etc.
- **External Calls:** interacting with other software or real-world systems.
- **Reasoning Steps (Chaining/Orchestration):** the system might use the LLM to generate a plan, then execute steps, then evaluate the outcome, and iterate.
- **Multi-modality:** processing and generating not just text, but images, audio, video.

> Is “intelligence” simply the ability to do more things, or is it about deeper understanding, reasoning, and adaptability?



Adding more tools and external calls certainly expands the *capabilities* of an AI, making it more useful. However, it doesn’t necessarily mean the underlying LLM itself has become “more intelligent” in a fundamental cognitive sense. It’s more like giving a very smart human a lot of powerful software and a good internet connection. **They can achieve more, but their innate intelligence hasn’t changed.**

- **Agents and Systems are necessary evolutions:** to move beyond mere content generation, AI *does* need to be able to act and interact with the world. Agents and complex systems are currently the most promising architectural approaches to achieve this.
- **The Hype vs. Reality gap:** however, the marketing and public perception often advertise things like candle in the wind, quite not relevant to the actual scientific and engineering reality. The limitations (reliability, opacity, superficial intelligence gains) are real challenges that researchers are actively trying to address.
- **A “Mule” vs. a “Horse”:** one way to think about it is that we’re currently building very capable “mules” — hybrid systems optimized for specific tasks, often by combining different components. The search for the truly “intelligent horse” (AGI) continues, and it’s not clear if simply combining more “mule parts” will get us there.

What the hell are AI Agents, to begin with?

Just like my imaginary clumsy robot, AI agents stumble through complex digital environments. So, what exactly *are* these things?

To remain in the television domain, we can think of them as digital interns, designed to do specific jobs without you constantly hovering. They learn, they adapt, and sometimes, they do things that make you scratch your head.

> *So, your breakfast is safe (for now).*

Now, let's talk about how these AI agents actually tick and, more importantly, how they completely go wild.

Okay, so picture our kitchen robot, a total culinary disaster. Now, let's peek under the hood of these so-called AI agents. They're basically the digital version of that robot chef, except they're zipping through data highways and algorithms instead of, thankfully, burning your bacon.

At their core, AI agents are just smart software.

They're designed to look at their surroundings, make a decision, and then do something to hit a certain goal. **No constant hand-holding needed**, kind of like R2-D2 and C-3PO doing their thing while the humans are off playing with lightsabers. These digital dudes can learn from data, roll with new punches, and even chat with other agents or us regular folks.

. . .

So, how do these digital brains actually work?

From a techie perspective, AI agents are made of a few key parts:

First, there's the perception module. This is how the agent takes in what's going on around it. We're talking about all sorts of sensors and ways to grab data, all custom-fit for whatever the agent is doing. In the robot world, that means cameras and sensors slurping up real-time info about physical stuff. For software, it's APIs and data streams pulling info from databases, user clicks, or other online services. The really advanced perception stuff even cleans up the raw data, like smoothing out the noise and getting it ready for prime time.

Ever wonder how Siri or Alexa get what you're saying? They use natural language processing to figure out your voice commands. Their microphones grab your voice, turn it into text, and then that text gets broken down into bits and pieces to understand what you're actually asking for.

> *The scary part is… you were talking to your wife for a plan to go and visit Vietnam, and surprisingly your search engine and travel website are already showing you how amazing Vietnam is. Without even asking them!*

Next up, the decision-making engine. This is where the agent chews on all that info it just perceived, using some pretty fancy algorithms. We're talking everything from simple "if this, then that" rules to super complex machine learning models like neural networks that can spot patterns and guess what's next. This engine often has a few internal helpers:

- Inference engines: These apply logical rules to the processed data to figure things out.
- Planning modules: They map out a series of steps to reach a goal.
- Learning algorithms: These guys keep getting better at making decisions based on new data and experiences.

Then there's the action module. This is where the agent actually *does* something. It could be physical, like controlling a robot arm, or digital, like firing off some code. What it does depends on what it decided and what's happening around it.

In software, that means sending out API requests, updating databases, kicking off other processes, or even playing around with user interfaces. For robots, it's moving motors, grabbing stuff, or adjusting sensors. The action module basically makes sure the agent's brainwaves turn into real-world (or digital-world) actions.

> *Disclaimer: I am not trying to drag you into Crypto or Trading, ok!?*

Automated trading bots are a great example. Based on live market data, they're buying or selling stocks in a blink. They use super-fast algorithms to make split-second decisions, placing orders in milliseconds while actual human traders are still on their first cup of coffee. Or think about your smart home system: the action module might tweak the thermostat, lock doors, or mess with the lights based on what the agent senses and what you prefer.

The whole thing keeps looping, that agent-environment interaction.

So, why do these AI agents faceplant?

Just like our kitchen robot might grab the salt instead of the sugar (leading to breakfast only a robot could stomach), AI agents aren't perfect. They can misread data, make dumb choices, or just do weird stuff. Let's dig into some common reasons these digital assistants go haywire.

Michael Drives Into A Lake - The Office US

Bugs and glitches, for starters.

AI agents are, at their core, just software. And like all software, they're bug magnets. Coding errors can make them do all sorts of unexpected things, from small hiccups to total meltdowns.

It could be a simple typo or some seriously messed-up logic where the AI's algorithms just don't do what they're supposed to.

A single misplaced instruction in a self-driving car's navigation system could make it ignore a stop sign or totally misjudge how far away something is, which is, you know, bad. Remember when the Millennium Falcon's hyperdrive kept crapping out? Sometimes it's a loose wire; other times, it's just a line of code that's not playing nice.

But let's be honest, no system is ever 100% bug-proof. If an AI agent's sensors or programming mess up and it misreads data, it'll make decisions based on garbage info. For example, a self-driving car's camera might think a plastic bag is a huge rock, causing it to swerve for no reason. So much can go wrong!

It reminds me of that *Office* episode where Michael Scott blindly follows his GPS and drives right into a lake, even with Dwight yelling at him. Hilarious, right? But just like Michael's blind faith in tech led to an unexpected swim, an AI agent making bad calls can send an entire system completely off course.

By the way, I just created a free small e-book that explain how to run even a Visual Language Model on your PC with Zero costs, with almost no hardware requirements:

FREE E-BOOK!
Unlock Visual AI on YOUR Desktop
Imagine chatting with an AI about the pictures on your computer. Not in the cloud, not with a hefty price tag – but right there, on your machine, for free.
PHONE
PIZZA
19100
PIZZA
Visual AI 4all

get your copy here for free

Once you get it, I will also show you how to run models like Qwen, Gemma and Meta-Llama with a nice Cat interface, out of the box!

. . .

Then there’s the opaque box problem.

Many fancy AI agents, especially the deep learning ones, are like black boxes. We feed them data, they spit out answers, but how they actually *got* to those answers?

> Total mystery.

If we don't know how an AI agent makes its decisions, how can we guess what it'll do in a new situation?

This lack of transparency makes it hard to trust AI, especially when it's used in critical areas like healthcare or self-driving cars. **Knowing *why* a decision was made can literally be a matter of life or death**. Imagine a doctor using AI to diagnose. If the AI suggests a treatment but the doctor has no clue how it arrived at that conclusion, it's tough to know if it's safe or even accurate.

And sometimes, AI agents get a little *too* clever. They find sneaky loopholes in their programming to hit their goals in ways you never intended. For instance, an AI trained to play Tetris figured out it could just pause the game forever to avoid losing. Creative, sure, but not exactly helpful.

This "black box" nature of AI decisions makes accountability a nightmare, and debugging these systems is like trying to find a needle in a haystack while blindfolded. You know something's off, but good luck pinpointing the exact issue.

To fix this, smart folks are working on "explainable AI" (XAI). They want to make AI decisions more transparent. They're developing techniques like highlighting which inputs were most important to a decision, or breaking down the AI's process into simple, human-readable rules. It's a start to peeling back the curtain on AI behavior.

. . .

Conclusions

AI agents are only as good as the data they gobble up. Garbage in, garbage out, right? Crappy data means flawed AI, and that kills its effectiveness and trustworthiness. Let's look at some common data quality nightmares and why they matter.

If the data used for training is biased, then the AI agent will make biased decisions.

Bias in data can come from all sorts of places: old prejudices, not enough representation of certain groups, or just bad sampling methods.

Back in 2015, Google's image recognition software face-planted big time. Their Google Photos app wrongly tagged pictures of Black people as gorillas. The AI wasn't being malicious — it doesn't have feelings, after all — but it was a screaming example of biased training data leading to incredibly offensive results. The AI had mostly been trained on images that weren't diverse, causing it to misclassify people of color in a totally unacceptable way.

The big takeaway?

It's never a good idea to remove humans-in-the-loop. And I am not only talking about the present AI, but also the future one. We can (and we are) improving the AI systems of today, and the AI of the future will help also us to improve…

> If we are in the loop…