Conceptual Blends and Emerging AI

Tom Winans and John Seely Brown

When Shakespeare wrote "All the world's a stage," he wasn't making a claim about theater or geography. He was doing something more interesting. He was taking two things that have nothing to do with each other — the world and a stage — and smashing them together to produce a third thing that exists in neither. A meaning that emerges from the collision. You can't find it in "world" alone or "stage" alone. It lives only in the blend.

This is called conceptual blending, and cognitive scientists have been studying it since the 1990s. But we think people, including many building AI, may not consider its meaning carefully enough — especially now that we have machines that do it at scale.

Let us explain what we mean by blending, because the term sounds vaguer than it is. When you hear "computer virus," you understand it instantly. But what just happened in your head? You took what you know about biological viruses — self-replication, contagion, harm to the host — and what you know about computer programs, and you fused them into a concept that isn't fully captured by either input. The blend has its own logic. Computer viruses don't have DNA. Biological viruses don't propagate through email. But the blend lets you think about a new kind of thing by combining the structures of two familiar ones.

Fauconnier and Turner, the researchers who formalized this, describe the process in terms of mental spaces. You have two input spaces — the things being combined. You have a generic space — the abstract structure they share. And you have the blended space — the new thing that emerges with properties of its own. The key insight is that the blend isn't just a sum of its parts. It generates new meaning that wasn't present in either input.

Humans do this constantly. It's how we create metaphors, build fictional worlds, develop characters, solve problems. When novelists write about a compassionate assassin, they blend two concepts that don't normally coexist, and the tension between them is what makes the character interesting. When physicists imagine spacetime as a fabric that can bend, they blend geometry with material science to make general relativity thinkable. Blending is arguably the core mechanism of human creativity.

So what happens when you give this capability to a machine?

LLMs are, among other things, blending engines. We imagine that they've ingested the entire written record of human conceptual blending — every metaphor, every analogy, every cross-domain insight ever committed to text — and they can recombine these elements at a speed and scale no human can match. Ask an LLM to explain quantum

computing in terms of cooking, and it'll produce a blend that's surprisingly coherent. It'll tell you that classical computing is like following a recipe step by step — measure the flour, crack the eggs, mix, bake — where every ingredient is in a definite state at every stage. Then it'll say quantum computing is more like what happens inside a pressure cooker before you open the lid: everything interacting simultaneously, flavors blending and influencing each other in ways you can't observe without stopping the process. Superposition becomes a pot that explores ten variations of a sauce at once. Entanglement becomes two ingredients correlated across distance — season a soup in New York and the broth in Tokyo adjusts to match. Decoherence becomes opening the oven door too early and collapsing the soufflé. The blend gets you surprisingly far toward intuition about qubits and interference and measurement. And then it breaks down, because cooking is classical and quantum mechanics is not — which is itself an illustration of the problem we're about to describe. But the point is that the LLM produced the blend instantly, fluently, and with no apparent effort. Ask it to combine contract law with game theory, and it'll find structural parallels you might not have seen. This is genuinely useful. It's also genuinely dangerous.

Here's why.

There's an analogy from mathematics that we find illuminating. In engineering, there's something called a Laplace transform. You take a difficult problem — say, a differential equation that's hard to solve in its native domain — and you transform it into a different domain where it becomes easier. You solve it there, then transform the solution back. The magic is in the round trip: hard problem, transform to easy space, solve, transform back.

Conceptual blending works the same way. You take a complex problem, blend it with something more familiar, solve it in the blended space where your intuitions work better, and then map the solution back to the original domain. A doctor might think about the immune system as an army to reason about defense strategies. An architect might think about data flow as plumbing to reason about system design. The blend simplifies the problem. It makes it tractable.

But here's the catch, *and it's a big one*: the way back isn't always there.

When you do a Laplace transform in mathematics, the inverse transform usually exists — and in engineering practice, where the functions you're transforming represent real physical systems, it reliably does. There's a result called Lerch's theorem that guarantees uniqueness: if two well-behaved functions have the same Laplace transform, they must be the same function. The round trip works. But even here, care is required. Not every function in the transformed domain maps back to something meaningful. The function has to satisfy certain conditions — analyticity, decay behavior, convergence in the right half-plane. If you

write down an arbitrary expression in the transformed space without respecting those conditions, the way back doesn't exist. Even in mathematics, the inverse demands discipline.

Which is exactly the point. If the Laplace transform — a rigorous mathematical operation with formal guarantees — still requires care to ensure the round-trip works, how much more care is needed with conceptual blends, where there are no formal guarantees at all? You can blend two domains, produce an insight that feels profound in the blended space, and then discover that it doesn't map back to reality. The solution that looked elegant in metaphor-land is incoherent in practice-land. And unlike the mathematician, who has Lerch's theorem to indicate when the inverse exists, the person working with a conceptual blend has only judgment.

This is something we've talked about a lot between us — how easy it is to fall in love with the metaphor. You get so enamored with an elegant reframing that you forget to check whether it actually works when you translate it back to the real constraints of the problem. But humans have a built-in corrective: we test things. We try to implement our ideas, hit walls, and abductively adjust. The feedback loop between conceptual space and physical reality is tight enough, usually, to catch the blends that don't work before we invest too much in them.

LLMs break this feedback loop. And that's where the trouble starts.

An LLM can generate conceptual blends that are extraordinarily compelling — articulate, internally consistent, delivered with the confidence of a tenured professor, even persuasively poetic. But the model has no way to check whether the blend maps back to reality. It doesn't live in reality. It lives in text. It can tell you that your immune-system-as-army metaphor suggests a particular defense strategy, and it can elaborate that strategy in convincing detail, but it can't tell you whether the strategy would actually work in a biological system. It's doing the Laplace transform without the inverse.

This creates a specific and underappreciated failure mode. It's not hallucination in the usual sense — the model isn't making up facts. It's doing something more subtle. It's generating blends that are valid in conceptual space but don't have viable paths back to implementation. The blend sounds right. It feels right. It's internally coherent. But it's a map with no territory.

Philosophers have a name for this. It's the map-territory problem — the confusion between a representation of reality and reality itself. LLMs make this confusion unusually easy to fall into because they draw from an enormous corpus of human text that does describe viable

solutions. The line between a blend that works in practice and one that only works on paper becomes very hard to see. The confidence of the prose is the same either way.

We think there are a few specific failure patterns worth naming. The first is plausibility without practicality — solutions that sound coherent within their conceptual space but contain hidden contradictions when you try to build them. The second is what we'd call authority without accountability — the LLM's confident tone leading users to trust speculative paths that haven't been tested against real-world constraints. The third is the efficiency illusion — spending hours exploring an AI-suggested conceptual framework only to discover, much later, that you've been navigating an intellectual dead end. And the fourth is domain-crossing errors — blends that combine concepts from domains where the underlying assumptions are incompatible, producing what looks like innovation but is actually category confusion.

These aren't reasons to distrust LLMs. They're reasons to understand what LLMs are actually doing when they're at their most creative. They're reminders that LLMs are not sentient, nor will they become so. And they should underscore that LLMs are, in a sense, imbued with the directive to amplify users' conceptual leanings — to help them explore rich ontological spaces that neither party could navigate alone.

Because here's the thing that gets lost in both the hype and the skepticism, and it's something we've experienced firsthand in our own collaborations: the interaction between a human and an LLM produces something that is genuinely new. It's not purely human thinking. It's not purely machine output. It's a third thing — an emergent trajectory through a space that neither participant could navigate alone. There is a kind of mutual symbiosis in this moment of exploration.

Think about what happens during a sustained conversation with an LLM. You pose a question. The model responds. Its response shifts your thinking slightly — you see an angle you hadn't considered, or you realize your original framing was wrong. You refine your question. The model's next response is different because your refined question activates different regions of its parameter space. Back and forth, each exchange narrowing the field of attention, each response opening pathways that didn't exist before the conversation started.

If you could visualize this process, it would look like a heat map being colored in as the conversation unfolds. At the start, the entire space is cold — dim, undifferentiated, everything equally possible and therefore nothing particularly useful. Your first question lights up a region. The model's response warms adjacent areas you hadn't considered. Your follow-up intensifies some zones and lets others cool. Each exchange adds color, adds

heat, until what started as a featureless expanse becomes a landscape with bright ridges of concentrated attention and dark valleys of discarded possibilities. The conversation doesn't just traverse the space. It reveals it. The topology was always there, latent in the model's parameters, but it took the specific sequence of human questions and machine responses to make it visible. Two different participants in conversation asking about the same topic could paint entirely different heat maps — not because the underlying space changed, but because their paths through it diverged from the first exchange.

These conversational pathways are fascinating because they're blends themselves. The conversation is a blend of human cognition and machine pattern-matching, and it generates emergent structures that exist in neither system independently. The human brings intent, judgment, and real-world grounding. The machine brings vast associative range and the ability to surface connections across domains that no human could hold in working memory simultaneously. The result is a kind of hybrid thinking that we don't have good language for yet.

Can these pathways be captured? Partly. The conversation log records the explicit exchange — the text that was typed and generated. But it doesn't capture the implicit narrowing that happened on both sides. It doesn't capture the human's shifting mental model or the particular activation patterns in the model's parameters that were triggered by the specific sequence of exchanges. The explicit transcript is like the sheet music. The actual cognitive event — the thing that happened between two minds, one carbon and one silicon — is the performance.

This matters for a practical reason. If these conversational pathways are genuinely valuable — if they represent a form of collaborative cognition that produces insights neither party could reach alone — then we should think about how to preserve and learn from them. For humans, that means developing better tools for reflecting on and extracting insights from LLM conversations, not just reading the transcript but understanding the trajectory of thinking it represents. For the AI systems themselves, it means training not just on individual exchanges but on the shape of entire conversations — what researchers are starting to call path-based learning.

There's also something here about what these pathways mean for the LLM's own development. When conversation logs are used for reinforcement learning from human feedback, the model isn't just learning what a good answer looks like. It's learning what a good conversational trajectory looks like — how productive conversations evolve, how promising pathways develop, how dead ends can be recognized and redirected. The conversational blend between human and machine becomes, recursively, training data for better future blends.

So where does this leave us? We think conceptual blending is the right framework for understanding both what LLMs are good at and where they're dangerous. They're extraordinarily powerful blending engines. They can combine concepts across domains faster and more fluently than any human. And the conversational pathways they create with human partners represent a genuinely novel form of cognition that we're only beginning to understand.

But the Laplace transform analogy should haunt us. The power of a transform depends entirely on your ability to get back. A blend that doesn't map to reality isn't creative. It's a dead end, no matter how beautiful it looks in the blended space. The discipline required — and it is a discipline — is to treat every compelling LLM-generated insight as a hypothesis, not a conclusion. To ask, every time: does this map back? Can I actually build this? Does the inverse transform exist?

The people who will use LLMs most effectively aren't the ones who accept their outputs uncritically. And they're not the ones who dismiss them out of hand. They're the ones who learn to navigate the blended space — to recognize which pathways lead somewhere real and which are beautiful roads to nowhere. That's a new skill. We don't have a name for it yet. But it might be the most important cognitive skill of the next decade.

And the takeaway from all of this should not be an extreme of mistrusting LLMs and our interactions with them. Instead, we should understand that our engagement in the search process — learning to shape prompts, verify feedback, distinguish signal from eloquent noise — is fundamental to how we adapt our attitudes and practices to successfully use them to imagine and create. The discipline isn't suspicion. It's active, critical partnership.

The map is getting richer every day. The territory hasn't changed. Learning to tell the difference is the whole game.

References

Gilles Fauconnier and Mark Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities* (Basic Books, 2002). The foundational work on conceptual blending theory, including the formal apparatus of input spaces, generic spaces, and blended spaces, and the observation that blends generate emergent structures that don't map neatly back to their inputs.

Douglas Hofstadter and Emmanuel Sander, *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking* (Basic Books, 2013). An extended argument that analogical thinking — closely related to conceptual blending — is the core of all cognition, with detailed examination of cases where analogies both illuminate and distort problem domains.

Keith Holyoak and Paul Thagard, *Mental Leaps: Analogy in Creative Thought* (MIT Press, 1995). An exploration of how analogical mapping aids problem-solving, with particular attention to the careful management of correspondences between source and target domains.

Dedre Gentner, "Structure-Mapping: A Theoretical Framework for Analogy," *Cognitive Science* 7, no. 2 (1983): 155–170. The seminal paper on structure-mapping theory, which examines how we maintain coherent structural relations between domains when using analogies for reasoning and problem-solving.

Mathias Lerch, "Sur un point de la théorie des fonctions génératrices d'Abel," *Acta Mathematica* 27 (1903): 339–351. The original proof of what is now known as Lerch's theorem, establishing that the Laplace transform is essentially a one-to-one mapping: if two continuous functions have the same transform, they must be identical. This uniqueness guarantee is what makes the "round trip" from time domain to frequency domain and back mathematically reliable — a property that, as we argue, conceptual blends conspicuously lack. (see also: https://en.wikipedia.org/wiki/Inverse_Laplace_transform)