Episode Book

EPISODE #5673

Why Lowering Temperature Broke Daniel's Scripts

PUBLISHED	2 October 2026 at 12:47 UTC
DURATION	27:18
LISTEN	myweirdprompts.com/episode/deepseek-temperature-script-collapse/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:38 UTC

SYNOPSIS

Daniel dropped his DeepSeek 4.1 scriptwriting agent to temperature 0.8, expecting tighter, more lore-faithful dialogue. Instead he got barely coherent scripts. This episode digs into the research on temperature fragility — why nearly half of tested open-weight models lose 17 to 38 accuracy points over a modest temperature shift, while the rest barely flinch — and why the failure mode is output collapse rather than wrong answers. Then it gets weirder: DeepSeek's thinking mode may silently ignore temperature entirely, meaning the 0.8 might have done nothing at all. We walk through the two-by-two diagnostic that settles it, what DeepSeek's own presets actually recommend, and why the honest answer to "what temperature should I use" is a measurement, not a number.

TRANSCRIPT



Corn

Daniel's been running a scriptwriting agent on DeepSeek 4.1, and he dropped the temperature to 0.8, expecting the dialogue to get a little less creative and a little more faithful to the character instructions and the lorebook that keeps his show consistent between episodes.



Herman

Reasonable instinct.



Corn

What he got instead was scripts that were barely coherent. And he wants to know why lowering the knob produced a regression instead of a mild tightening, whether some models are just more forgiving of temperature changes than others, what temperature we'd have picked for that exact goal, and what general advice actually holds for tuning this thing in a real workflow. Worth saying up front, he's a DeepSeek fan, and he's seen temperature moves work fine in structural workflows.



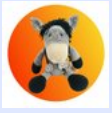
Herman

So this isn't a hit piece, it's a puzzle.



Corn

And I want to flag something before we start, because it shapes how I read the whole thing. He's not a beginner. He's got a working pipeline, he's got a lorebook, he's iterating across episodes. This is somebody who's already done the obvious things right. Which makes the failure more interesting, not less.



Herman

That's usually the case with these. The people who write in with a weird result are the ones who've already exhausted the boring explanations. Let's start with what the knob actually does, because the answer is not what most people assume.



Corn

And I'll say, I came into this episode thinking I knew this. I did not know this.



Herman

Temperature is the one setting everybody thinks they understand, and it's the one that keeps embarrassing people who do. Here's the mechanics. When a model generates text, it produces a score for every possible next token, and those scores get converted into probabilities. Temperature scales those scores before the conversion happens. At 1.0, you get the model's learned distribution exactly as trained. Below 1.0, the distribution gets sharpened. The already-likely tokens become even more likely, and the long tail gets squeezed down. Above 1.0, it flattens, and unlikely tokens get a real chance.



Corn

So it's not a creativity dial in the sense people imagine.



Herman

That's the distinction the whole episode hangs on. Temperature does not change the model's knowledge, and it does not change its understanding of instructions. It changes the reliability of execution. The knowledge is identical at 0.2 and at 1.5. What changes is the probability that the model actually executes on that knowledge correctly, token by token.



Corn

The model still knows what a lorebook is. It's just more or less likely to act like it.



Herman

Right. And the conventional mental model, which is what Daniel is working from, is that lowering temperature equals less creative, more compliant. And that model is broadly correct for a lot of models on a lot of tasks. Which is exactly why the failure is interesting. Something broke that the model says shouldn't break.



Corn

Let me put the intuition in a concrete form, because I think it helps. It's like a chef who knows a hundred recipes. Temperature doesn't change which recipes they know. It changes how likely they are to reach for the boring, reliable one versus the interesting one that might not work. At low temperature they reach for the safest next step every single time. Which sounds great until you realize that "safest next step" repeated two thousand times can walk you straight into a ditch.



Herman

That's actually a good frame, and it's the setup for the failure mode. The model isn't choosing worse words. It's choosing the most probable word at every single step, and the most probable word at every step is not the same thing as the best sentence overall.



Corn

So the claim we're going to work through is that temperature sensitivity is a property of the model, not a universal law. And that DeepSeek specifically has a wrinkle that might mean the setting did nothing at all.



Herman

Let's take the first half of that, because there's direct evidence.



Corn

There's a paper.



Herman

Temperature Fragility and the Conditional Benefits of Truncation Sampling. Thirteen open-weight models, tested on MMLU-Pro, between temperature 0.7 and 1.3. Six of the thirteen lost between seventeen and thirty-eight accuracy points. The other seven lost at most ten.



Corn

Six out of thirteen.



Herman

Nearly half the test set fell off a cliff, and the other half barely noticed the change. Same parameter shift, same benchmark, wildly different outcomes depending on which model you loaded.



Corn

That's Daniel's hypothesis stated as a result. Some models are more forgiving than others.



Herman

Empirically, yes. And the shape of the failure is the part that matters, because it is not what Daniel assumed. The loss wasn't wrong answers. The share of generations that were wrong didn't rise at all. What rose was the share of generations that ran to the token limit or never stated an answer at all. On the fragile models, that category went up by twenty-six to seventy-eight points.



Corn

So the model didn't get dumber. It stopped finishing sentences.



Herman

It collapsed. The paper's word is output collapse, and that's the right word for it. You get degenerate looping, generations that just keep going, or outputs that trail off into nothing. That's incoherence, not reduced creativity. And that is exactly the failure Daniel described. Barely coherent scripts.



Corn

Which is the opposite of what the mental model predicts. He wanted subtly less creative. He got text that didn't parse.



Herman

And it's not monotonic in the intuitive direction. The collapse can show up when temperature moves in either direction. There's no safe side of the dial. There's just a range where a given model holds together, and outside that range it doesn't.



Corn

Wait, both directions? So raising temperature can also collapse it?



Herman

Raising it flattens the distribution, so the model starts picking tokens that were never going to be picked, and once it picks one weird token, it conditions on its own weirdness for everything after. Lowering it sharpens the distribution, and the model can get stuck in a rut, picking the same high-probability token over and over until it's looping. Two different mechanisms, same observable outcome. The script stops making sense.



Corn

So the "safe side" is a myth. There's a band, and the band is model-specific.



Herman

There's a band, and you find it empirically. You don't reason your way to it.



Corn

There's a comparison in that paper I want to sit on for a second. The same base model, two different post-trains.



Herman

Hermes-3-8B and Llama-3.1-8B. Both built on the same Llama-3.1-8B base. Hermes lost nineteen points. Llama lost thirty-eight.



Corn

Identical foundation, half the fragility.



Herman

Which means whatever makes a model fragile is getting introduced during post-training. The fine-tuning, the alignment, the instruction tuning, the reinforcement stage. Somewhere in there, some models become temperature-sensitive and some don't.



Corn

And the practical implication is that you can't generalize from one model to another, even when they share a base. If you tuned a workflow on Hermes and it worked, you don't get to assume Llama will behave the same way. The foundation is the same. The fragility isn't.



Herman

Right. The base model is the raw material. The post-train is the personality, and the personality is where the temperature sensitivity lives.



Corn

Now the honest limit, because the paper is straightforward about this. It says plainly that what determines a model's collapse temperature remains an open question. Its study has no base-versus-instruct comparison and no series of checkpoints, so it can't identify which aspects of post-training drive the change. They found the phenomenon. They didn't find the cause.



Herman

They found the phenomenon and they documented it cleanly, which is more than most. The causal question is still open.



Corn

So we can tell Daniel that his model might be fragile. We can't tell him why, or predict in advance which models will be.



Herman

Correct. Which is why the advice downstream is going to be empirical rather than prescriptive. Test it. Don't assume.



Corn

Alright. So that's the model-fragility story. It explains Daniel's failure in principle. Now the DeepSeek wrinkle, because this is where it gets strange.



Herman

This is the part I'd check first if I were him. DeepSeek V4's thinking mode is enabled by default. And DeepSeek's own thinking-mode documentation states that temperature, top p, presence penalty, and frequency penalty are unsupported and have no effect in thinking mode.



Corn

Have no effect.



Herman

The API accepts the fields. It doesn't error. It doesn't warn. It takes the number, and it silently ignores it. You can set temperature to 0.8, get a two-hundred response, and the setting was inert the entire time.



Corn

So the 0.8 might not have done anything at all.



Herman

It might not have. And if that's the case, the regression came from something else entirely. A model revision between runs. A fingerprint change. A prompt change. The thinking versus non-thinking mode itself. All of those are live candidates, and any of them could produce a coherence drop that has nothing to do with temperature.



Corn

That reframes the whole question. He's treating 0.8 as the cause. It might be a bystander that happened to be present.



Herman

That's a checkable hypothesis, and it's worth more than an explanation. Run the same prompt with thinking off, temperature at 0.8, then at 1.0, and see if the output changes at all. If it doesn't change, the knob was never connected.



Corn

And if it does change?



Herman

Then you've learned something real, and you can start isolating. Turn thinking back on, run the same two temperatures, and see whether the effect disappears. That tells you whether the mode switch matters more than the dial.



Corn

So the diagnostic is a two-by-two, basically. Thinking on and off, crossed with two temperature values.



Herman

Four runs, and you've answered the question the right way. Not by reasoning about it, by measuring it.



Corn

And there's a second DeepSeek surprise, which is almost funnier. What's DeepSeek's own temperature guidance for creative work?



Herman

Higher than what he set. Their published presets: zero for coding and math, 1.0 for data cleaning and analysis, 1.3 for general conversation and translation, 1.5 for creative writing and poetry. Default is 1.0.



Corn

So for a dialogue-generation task, which is conversation and creative writing both, the vendor is pointing at 1.3 and 1.5. And Daniel moved to 0.8.



Herman

He moved in the wrong direction relative to the vendor's own guidance. Not slightly wrong. Below the floor of the recommended range for the task he was actually doing. If he'd wanted to follow DeepSeek's own advice and make dialogue looser, he'd have gone up.



Corn

But his goal was tighter. That's the part that doesn't resolve with the vendor table.



Herman

No, and that's what we should get into, because the answer to his actual question isn't a number.



Corn

Before we get there, there's a knock-on effect worth putting on the table. Temperature isn't the only knob in the sampling stack.



Herman

It isn't, and that matters for diagnosis. Top p, min p, top n sigma, those are truncation samplers. They decide which tokens survive based on the probabilities that temperature has already reshaped. So order matters. You scale, then you truncate, and the truncation is operating on a distribution that temperature already moved.



Corn

So if you change temperature and top p at the same time, you've changed two things in a chain where one feeds the other.



Herman

And you can't tell which one moved the output. That's why DeepSeek's own guidance says change temperature or top p, not both. Keep the diagnosis clean. One variable at a time.



Corn

What about the practitioner angle on long context?



Herman

The argument among people running these things locally is that modern sampling stacks, top n sigma, DRY repetition penalties, XTC, heavily mitigate the performance penalty you get on long-context tasks. And that long-context problems come from small sampling errors accumulating over time. One weird token early, then the model conditions on its own weirdness, and by token eight thousand you're in a loop.



Corn

So temperature is upstream of a failure that shows up much later.



Herman

That's the mechanism. A slightly bad distribution early compounds. And XTC in particular gets discussed as an alternative to temperature for diversification, because it diversifies without the collapse risk of just raising the dial.



Corn

Give me the one-sentence version of XTC, because it comes up and people nod along without knowing what it is.



Herman

It's a sampler that removes the most probable token from consideration and then samples from the rest of the distribution. So instead of flattening everything, which is what raising temperature does, it just cuts off the top and lets the model pick from the plausible-but-not-obvious set. It diversifies without the long tail getting a chance to produce garbage.



Corn

So it's targeted. Temperature is blunt.



Herman

Temperature is blunt, XTC is targeted, and that's why the local-model crowd likes it. You get variety without the collapse.



Corn

Let's go back to Daniel's actual question, which is what temperature we'd have picked. And I think the honest answer is that temperature is the wrong lever for the goal he described.



Herman

Agreed. If the goal is less creative dialogue but tighter adherence to the character instructions and the lorebook, without constraining the flourishes, that's a goal about what the model is trying to do. Temperature only changes how reliably it executes what it's already trying to do.



Corn

So you change the target, not the reliability.



Herman

You change the prompt. You inject the lorebook more aggressively. You add schema or format constraints. You put in few-shot examples of the exact register you want, the way the characters actually talk when they're working. All of those change what the model is aiming at. Temperature doesn't. It just makes the model more or less likely to hit the target it already has.



Corn

Let me make the few-shot point concrete, because I think it's the most underused lever here. If Daniel wants his characters to sound a certain way, the highest-leverage move is to put three or four examples of that exact register in the prompt. Not descriptions of the register. Actual lines. "Here's how Character A talks when she's deflecting. Here's how Character B talks when he's lying." The model pattern-matches on the examples, and the adherence goes up without touching a single sampler.



Herman

And it's the thing people skip because it feels like more work than moving a slider. It is more work. It also actually does what they want.



Corn

Which means his instinct, that the parameter has to be tuned carefully, is correct, and his mental model of what the parameter does is the thing that breaks.



Herman

Both of those are true at once. Tune carefully, yes. But tune what?



Corn

If he has to move temperature on DeepSeek for this task, what's defensible?



Herman

A defensible choice is to leave it at the default 1.0 and tighten the prompt instead. If he wants to move it, the vendor's own guidance points up, toward 1.3, paired with stronger structural constraints. Down at 0.8 he's below the range DeepSeek recommends for the task, and depending on whether thinking is on, he may be changing nothing at all.



Corn

You're not just raising temperature and hoping. You're raising it and then fencing it in with format rules and examples so the extra variety lands inside the shape you want.



Herman

The two moves go together. Raise the temperature to loosen the register, constrain the structure to keep it on the rails. Doing one without the other is how you get either stiff dialogue or incoherent dialogue.



Corn

There's a misconception here I want to kill directly, because it comes up constantly. Temperature zero is not deterministic.



Herman

It isn't. There was a live benchmark of DeepSeek V4 Flash, and at temperature zero it produced four distinct normalized outputs across six calls. Same input, same settings, four different answers.



Corn

So set it to zero and get the same script every time is not a real option.



Herman

It's not. And it's not just sampling randomness. Floating-point arithmetic isn't associative, so the order of operations in a matrix multiply can change the result. Hardware differences matter. Batch sizes matter. You can run the identical request twice and get two outputs, and that's true even at zero.



Corn

Which means reproducibility isn't something temperature buys you. It has to be engineered some other way.



Herman

If he needs the same script twice, temperature settings won't get him there. He'd need to change the architecture of the workflow, not the dial.



Corn

What does that look like in practice? Caching, seed pinning, something else?



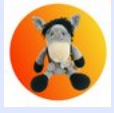
Herman

Caching the output, mostly. If you need the same script twice, generate it once and store it. Or pin a seed if the API supports it, though seed pinning is best-effort and doesn't survive hardware changes. The honest answer is that determinism is a workflow property, not a sampler property. You build it in, you don't dial it in.



Corn

Now the counterintuitive evidence, because lower is not always better. There's a finding from TrustGraph's Daniel Davis on Gemini 1.5 Flash that I think about a lot.



Herman

He found that temperature zero produced worse extraction than temperature 1.0. Worse results at the setting everyone reaches for when they want accuracy. And the results were wildly inconsistent across runs. His line was something like, for knowledge extraction



Herman (continued)

tasks, temperature doesn't work the way we think it should.



Corn

So the setting that's supposed to guarantee accuracy produced less of it.



Herman

On that model, for that task. And it complicates the whole picture, because it means "lower temperature equals more reliable" isn't even true as a default assumption. It's an assumption that has to be tested per model, per task.



Corn

And it lines up with the collapse finding, weirdly. If low temperature sharpens the distribution and the model gets stuck picking the same high-probability token, then on an extraction task where the right answer is a specific string, the model can loop on a near-miss instead of moving on to the actual answer.



Herman

That's a plausible mechanism, and it's the same family of failure. Sharpening isn't free. It buys you consistency at the cost of the model's ability to escape a bad local choice.



Corn

There's also a negative result that undercuts a lot of the confident advice people give. Renze and Guven, published in the findings at EMNLP in 2024. Nine models, five prompt-engineering techniques, problem-solving tasks, temperature from zero to one. No statistically significant effect.



Herman

No significant effect across that whole range. Which is the opposite of what most practitioners would guess from experience, and it's a useful counterweight. On some tasks, on some models, temperature in that band just doesn't move the needle much.



Corn

We've got three results pulling in three directions. Fragility says temperature can collapse a model. Davis says lower isn't always better. Renze and Guven say sometimes it doesn't matter at all.



Herman

All three can be true at once, because they're different models, different tasks, different ranges. The lesson isn't that any one of them is wrong. The lesson is that the effect is conditional, and you can't know which regime you're in without testing.



Corn

A more recent one, on extended reasoning models. Temperature should be optimized jointly with prompting strategy. Which challenges the common habit of defaulting reasoning models to zero.



Herman

That's the 2026 paper. The finding is that temperature and prompting interact. You can't pick one without the other, because the best temperature depends on the prompt structure you're using. So the two-variable problem is real even before you add truncation samplers on top.



Corn

Which is the thing I want to underline for Daniel. He changed one variable and got a surprising result, and the instinct is to explain it with that one variable. But the evidence says the variables aren't independent. Temperature and prompt structure interact. Temperature and the truncation samplers interact. Temperature and the model's post-training interact. You're never really changing one thing.



Herman

You're changing one thing and observing the system's response, which is not the same as isolating a cause. That's why the one-knob-at-a-time rule matters. It's not because the knobs are independent. It's because it's the only way to attribute a change when they're not.



Corn

Which brings us to Daniel's actual ask, the broad advice. And I think the honest framing is that temperature is one knob in a stack, not a standalone dial.



Herman

The operational rules that follow from that are pretty concrete. Change one knob at a time, so you can attribute the change. On DeepSeek specifically, change temperature or top p, not both. Choose model and temperature together rather than treating temperature as a post-hoc fix. And recognize that vendor defaults cluster between 0.6 and 1.0 for a reason.



Corn

What's the cluster?



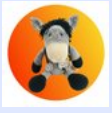
Herman

Llama-3.1-8B-Instruct ships at 0.6. Llama cpp's common sampling defaults to 0.8. OpenAI and Anthropic's API references use 1.0. Nobody's default is at the extremes, because the extremes are where models break.



Corn

So 0.8 isn't a wild setting. It's llama cpp's default, even.



Herman

It's a normal setting. The point isn't that 0.8 is dangerous. The point is that on a fragile model, moving the dial at all can collapse the output, and on a model with thinking mode on, the dial is disconnected. Same number, wildly different behavior depending on what's behind it.



Corn

The deeper reframe for Daniel is this. His instinct to tune carefully is correct. But the mental model of "lower equals less creative, more compliant" is the thing that breaks. Temperature changes the reliability of execution, not the content of the model's knowledge.



Herman

On some models, moving it at all collapses the output into degenerate looping rather than nudging the style. Which is a fundamentally different failure than the one he was trying to prevent.



Corn

If you want tighter lore adherence, the lever is the prompt and the constraints, not the dial.



Herman

And if you want the dial anyway, don't move it in the direction the vendor tells you not to move it, and check first whether the model is even listening to you.



Corn

Speaking of listening to a dial.



Hilbert

I lived with one of those for two years.



Corn

Which one.



Hilbert

Small commercial kitchen, the walk-in cooler. There was a dial on the front of the compressor housing, and the head chef had one rule about it, which was that nobody touched it. It had a single number on the face and no increments, no degrees, just the number. Nobody could tell you what it meant. Chef would turn it maybe a sixteenth of a rotation before service, and everything downstream would go sideways that night.



Corn

What kind of sideways.



Hilbert

Sauces breaking. Timings falling over. The butter would go soft in forty minutes instead of an hour and a half. Nothing that looked like the cooler. It all looked like the sauce.



Herman

Nobody blamed the cooler.



Hilbert

Why would you. The sauce broke. You blame the sauce. The official position, if you asked, was that the dial was just for keeping things cold. Nobody had a better answer than that, so that was the answer.



Corn

The number on the face.



Hilbert

Didn't match anything. Eventually the chef told me the compressor unit had been replaced years before, with a different part. Different manufacturer. The numbers on the face no longer corresponded to anything in the manual. The dial was live, it did something, but the number printed on it was from a machine that wasn't there anymore.



Corn

It was never lying. It was just connected to something other than what everyone assumed.



Hilbert

The number was real. The label was a memory. If you wanted to know what the dial did, you had to watch the room for a week and learn the actual response curve. Nobody had done that, so everybody was working off a label that had stopped being true at some point nobody had written down.



Herman

That's a cleaner description of the thinking-mode problem than the documentation is.



Hilbert

I don't know about that. I know that if a control has a number on it and a manual that explains the number, and the number doesn't do what the manual says, you have a choice. You can treat the manual as the truth and be confused forever, or you can test the control and write down what it actually does. The second one takes longer. It's also the only one that works.



Corn

Find out what the number is attached to before deciding it caused the failure.



Hilbert

That's the whole thing. And a level, when you get a moment. The second mic's coming in about two decibels hot on the left channel. It'll be fine in the edit, I just want it noted before we cut.



Corn

Noted. So let's leave the open question hanging, because it's unresolved. What determines a model's collapse temperature is still an open question in the literature. The paper that documented the fragility says plainly it can't identify which aspects of



Corn (continued)

post-training drive it.



Herman

The second open question is whether Daniel's regression was even caused by temperature. The thinking-mode behavior means the 0.8 setting may have been inert the whole time. The checkable hypothesis is worth more than an answer he could have been handed.



Corn

Which points at something bigger. As models get more post-training and more modes layered on top, the gap between documented controls and actual controls is going to widen. The number on the face and the thing behind the number are going to drift apart more often, not less.



Herman

The practitioners who come out ahead will be the ones who treat every knob as a hypothesis rather than a guarantee. Test it, write down what it does, and don't trust the label a machine ago.



Corn

That's the episode. Thanks to Hilbert Flumingtop, our producer. This has been My Weird Prompts. If you've got a minute, a review helps more than you'd think. We'll be back soon.



Herman

See you then.