# RTX 3090 Power Limit Tested: 280 W Costs 12% of Decode, 320 W Is the Sweet Spot

November 26, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** A popular YouTube tip says capping a 3090 at 280 W saves about 70 W for under 1% of speed. I measured it on a dense 27B. Against the 3090's 350 W reference limit, the 70 W saving is real, but decode slows by 12%, not 1%. Before testing, I predicted decode would barely slow down, since it's limited by memory speed rather than core power. That held at 320 W and failed below 300 W. Here's every step from 420 W down to 250 W, why decode falls off a cliff below 300 W, and the setting I'd use: 320 W.

📚 **More on this topic:** Used RTX 3090 Buying Guide · Fix Slow Qwen 3.6-27B on a 3090 · Multi-GPU Local AI

My 3090 pulls more than 400 W generating tokens from a 27B model, and the usual fix is a power cap: cap it at 280 W, save around 70 W, lose under 1% of speed. The version I tested comes from the YouTube video “I Tested Two $300 GPUs Against an RTX 5090”. The reasoning sounds right. Token generation is limited by memory bandwidth, not compute, so taking power away from the cores shouldn’t hurt much.

I measured it. My 3090 is a partner card whose default limit is 420 W, not the 350 W of the reference board. I ran it at 420, 350, 320, 280 and 250 W, then back at 420 W to check nothing had drifted. The model was Qwen3.8-27B UD-Q4_K_XL, a dense 27B, fully on the card with llama.cpp. The test was pre-registered, and the raw logs are public.

The 70 W is real. The 1% isn’t. And my own prediction was wrong too.

## The numbers

Each figure is the median of 3 requests. Watts are the card’s own reading, sampled every 100 ms during each phase. Decode is generating 512 tokens after a 512-token prompt. Prefill is processing a 4,096-token prompt.

| Limit | Decode tok/s | Prefill tok/s | Watts, decode | Watts, prefill | Joules per decoded token | Core clock, decode |
|---|---|---|---|---|---|---|
| 420 W (our card’s default) | 42.00 | 1,503 | 417 | 378 | 9.92 | 1,795 MHz |

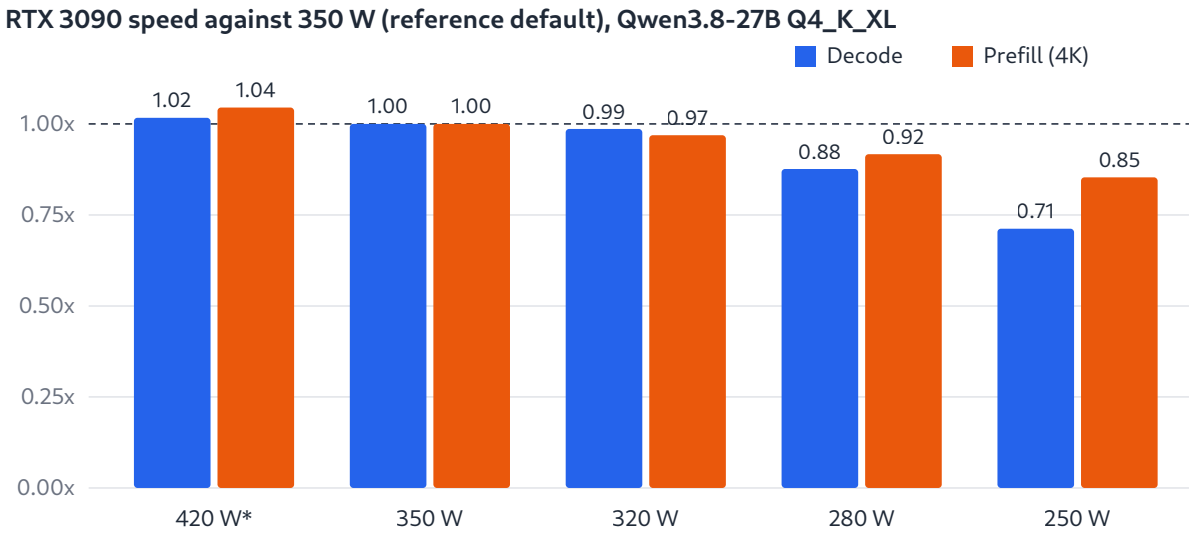
| Limit | Decode tok/s | Prefill tok/s | Watts, decode | Watts, prefill | Joules per decoded token | Core clock, decode |
|---|---|---|---|---|---|---|
| **350 W (reference 3090)** | **41.28** | **1,439** | 350 | 344 | 8.48 | 1,616 MHz |
| 320 W | 40.69 | 1,395 | 319 | 319 | 7.84 | 1,508 MHz |
| 280 W | 36.17 | 1,319 | 275 | 273 | 7.59 | 1,053 MHz |
| 250 W | 29.38 | 1,227 | 251 | 249 | 8.55 | 808 MHz |
| 420 W again | 41.95 | 1,506 | 418 | 361 | 9.97 | 1,803 MHz |

The same results as changes **against 350 W**, the reference 3090's default and the claim's baseline:

| Limit | Decode vs 350 W | Prefill vs 350 W | Joules per token vs 350 W | Watts saved, decode |
|---|---|---|---|---|
| 420 W (our card's default) | +1.7% | +4.5% | +17.0% | −67 W (draws more) |
| 320 W | −1.4% | −3.1% | −7.5% | 31 W |
| 280 W | **−12.4%** | −8.3% | −10.4% | 75 W |
| 250 W | −28.8% | −14.7% | +0.8% | 99 W |

If your card is a partner board with a higher default, like ours, here are the same steps **against our card's 420 W default**: 350 W −1.7% decode, 320 W −3.1%, 280 W −13.9%, 250 W −30.0%.

Inside each cell the three requests were within 0.3% of each other at 420, 350 and 320 W, so even the 1.4% step is real, not noise. The last row of the first table repeats the 420 W test after restoring the default: 0.999x the first run, so nothing drifted. The card never went above 72 °C.




Median of 3. *420 W is our partner card's own default. Memory clock 9,501 MHz at every limit.

## The claim, scored

The claim is 280 W against the 3090's 350 W reference limit, so that's the comparison:

| | 350 W | 280 W | Saved | Speed lost vs 350 W |
|---|---|---|---|---|
| Decode | 41.28 tok/s at 350 W | 36.17 tok/s at 275 W | 75 W | **12.4%** |
| Prefill | 1,439 tok/s at 344 W | 1,319 tok/s at 273 W | 71 W | **8.3%** |

The watts half is right: 70-75 W. The speed half is off by more than ten times. Against 350 W, you pay 8% on prompts and 12% on generation.

On a partner card like ours, against its 420 W default, 280 W saves 142 W during decode and costs 13.9%.

## My prediction was wrong too

Before testing, I predicted decode would barely slow down and prefill would take the hit. Prefill is heavy math, so it should feel a power cap. Decode on a single 3090 streams the whole model from VRAM for every token, so I expected it to be limited by memory speed rather than core power.

That held at 320 W: against 350 W, decode lost 1.4% and prefill 3.1%. It failed below 300 W, where the order flipped. Against 350 W, at 280 W decode lost 12.4% and prefill 8.3%; at 250 W

decode lost 28.8% and prefill 14.7%. Three of the predictions I wrote down before the run missed:

- Decode power at 420 W. I guessed 230-380 W. It was 417 W, pinned against the limit the whole time.
- Decode at 280 and 250 W, which fell below my bands.
- The shape. I said prefill would always lose more than decode, and it didn't.

## Why decode falls off a cliff below 300 W

The memory clock never moved. It read 9,501 MHz at every limit, 250 W included. What fell was the core clock: 1,616 MHz at 350 W, 1,508 at 320, then 1,053 at 280 and 808 at 250.

Here's my reading, and it's a hypothesis: I didn't measure memory power separately. The 3090's GDDR6X takes a big, fixed slice of board power when it runs at full clock, and decode keeps it at full clock. Under a cap, the driver protects the memory clock and takes everything out of the cores. Down to about 320 W the cores have power to spare, and decode barely notices. Below 300 W the cores get so little power that unpacking the 4-bit weights and doing the math takes longer than streaming them from memory. From then on, decode is limited by compute, and it falls faster than the power does.

It would also explain why decode sits on the power limit even at 420 W. Decode is limited by memory speed at full clocks, but that doesn't make it cheap on power.

## Where I'd set mine

**320 W.** Against 350 W it costs 1.4% of decode and 3.1% of prompt speed, saves 31 W while generating, and cuts energy per decoded token by 7.5% (7.84 J against 8.48).

On a partner card like ours, against its 420 W default, the same 320 W setting costs 3.1% of decode, saves 98 W and cuts energy per token by 21% (7.84 J against 9.92). That's where the setting really pays off.

280 W uses the least energy per token (7.59 J, 3.2% below 320 W), but going from 320 W to 280 W costs another 11% of decode speed. 250 W is worse than 280 W on both counts: slower, and more joules per token.

```
# See your card's limits first: current, default, min and max
nvidia-smi --query-gpu=power.limit,power.default_limit,power.min_limit,power.max_limit --format=
```

```
# Set 320 W (needs root, resets at reboot)
sudo nvidia-smi -pl 320

# Back to your card's default
sudo nvidia-smi -pl <default>   # the power.default_limit value printed above (350 on a referenc
```

One thing to watch: some 3090 advice, including one of my own guides, used to say `sudo nvidia-smi -pl 350` to "restore" the limit. On a partner card like mine, whose default is 420 W, that command lowers it. Check `power.default_limit` before you copy a number. nvidia-smi's documentation lists the power fields.

## What this doesn't cover

- One card, one model: a 420 W partner 3090 running a dense 27B with llama.cpp. An MoE model, which reads far fewer bytes per token, might behave differently. I planned to test one and didn't have the file on the box.
- Watts are the card's own reading, not a wall meter. The rest of the PC isn't counted.
- Three requests per cell. The spread inside each scored cell was under 2% at every limit, smaller than the steps at 280 and 250 W.
- No undervolting. A power limit and an undervolt aren't the same thing. An undervolt keeps clocks up at lower voltage and can do better than a plain cap. I only tested the cap.

The short version: against the 350 W reference limit, 320 W costs 1.4% and you'll barely notice. Going from 320 W down to 280 W saves another 44 W and costs another 11% of decode.

A weekly email with every new guide and measured benchmark.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/rtx-3090-power-limit-tested/

Free guides for running AI locally