disable Error-Correcting Code (ECC) memory on your **NVIDIA Quadro RTX 8000** in Ubuntu, run the `sudo nvidia-smi -e 0` command in your terminal and reboot your system. Disabling ECC frees up a portion of your 48GB VRAM that is otherwise reserved for parity bits and eliminates the minor memory bandwidth overhead caused by error-checking.

Step-by-Step Terminal Commands

Check your current ECC status. Run the following command to verify that ECC is currently enabled and note your GPU's index number (usually `0` if you only have one NVIDIA card):
bash

```
nvidia-smi -q -d ECC
```

Use code with caution.

Look for the **Current** status under the **ECC Mode** section.

Disable ECC mode. Run the command below with root privileges (use `-i 0` to target a specific GPU index if you have multiple cards installed):
bash

```
sudo nvidia-smi -i 0 -e 0
```

Use code with caution.

The terminal will confirm that the pending ECC state has been set to **Disabled**.

Reboot your machine. ECC changes only take effect after a full hardware reinitialization:
bash

```
sudo reboot
```

Use code with caution.

Verify the change. After Ubuntu boots back up, confirm that both the **Current** and **Pending** ECC states read `Disabled`:
bash

```
nvidia-smi -q -d ECC
```

Use code with caution.

Additional Notes

• **Re-enabling ECC.** If you ever want to turn ECC back on for scientific computing or mission-critical stability, run `sudo nvidia-smi -i 0 -e 1` and reboot.
• **Locking Boost Clocks.** Now that ECC is off, you can maximize sustained performance on Ubuntu by enabling persistence mode (`sudo nvidia-smi -pm 1`) and locking your graphics clock to its maximum boost frequency (`sudo nvidia-smi -lgc 1770` for a standard RTX 8000, or check your card's exact max clock with `nvidia-smi -q -d CLOCK`).