ChatGPT

Choosing a Higher-VRAM NVIDIA GPU on Runpod

Since the **RTX 4090 (24 GB VRAM)** might be a memory bottleneck for your workload, it's worth considering GPUs with **more VRAM** on Runpod. Below we outline several NVIDIA GPU options with larger memory, along with their specs, pricing, and how they might fit your needs. We also discuss whether you can reuse your existing code/optimizations on these GPUs or if any tweaks are needed. Finally, we address **on-demand vs. spot instances** on Runpod given your cost concerns.

NVIDIA GPU Options with More VRAM on Runpod

Runpod offers a wide range of GPU types. Here are some **GPUs that have more VRAM than the 24 GB RTX 4090**, along with their **specs and approximate pricing** on Runpod (hourly rate), and considerations for performance and compatibility:

- **NVIDIA** RTX 6000 Ada ***– 48 GB VRAM***: This is the professional "Ada Lovelace" GPU, essentially similar architecture to the RTX 4090 but with double the memory (48 GB). It delivers performance in the same ballpark as a 4090 (since it's the same generation) and would allow larger batch sizes or higher resolution without running out of memory. On Runpod, the RTX 6000 Ada is about **$0.40/hr** on-demand [1] (roughly twice the cost of a 4090 at $0.20/hr [2]). *Code/optimization compatibility:* No special changes needed – it uses NVIDIA's CUDA just like the 4090, so your existing PyTorch code and optimizations (mixed precision, etc.) will work out-of-the-box. This GPU is a great choice if you want **double VRAM with similar speed**, accepting a proportional increase in cost.

- **NVIDIA** RTX A6000 ***– 48 GB VRAM***: This is an Ampere-generation professional GPU (roughly analogous to a Quadro RTX A6000, similar compute to an RTX 3090 but with 48 GB memory). It also doubles your VRAM to 48 GB, but its raw compute performance is lower than the 4090 (Ampere vs. Ada architecture). The upside is **cost** – on Runpod an RTX A6000 is about **$0.25/hr** [3], only modestly more than a 4090. If your current 4090 was fully utilized (compute-wise) and just lacked memory, the A6000 might run somewhat slower per iteration (potentially ~50% of 4090's throughput, since Ampere has fewer CUDA cores and lower clocks). However, if the 4090 was **under-utilized due to memory constraints** (e.g. you had to use smaller batches or extra checkpointing that slowed things down), the A6000 could actually improve overall runtime by allowing a more efficient setup. *Code compatibility:* Also fully CUDA-compatible – no code changes required. This option is ideal if you **need more VRAM on a tighter budget**, and can tolerate some speed loss. (For reference, an **RTX A5000 (24 GB)** was cited as about half the price of A6000 for ~75% of the performance [4], which indicates the A6000's price/performance is a bit worse than mid-range cards – you mainly pay for the VRAM [5]. So choose A6000 if that extra VRAM is critical; otherwise, you'd stick with cheaper 24 GB cards.)

- **NVIDIA** A100 (80 GB) ***– 80 GB VRAM***: The A100 is a datacenter-grade GPU (Ampere architecture) with **80 GB of high-bandwidth memory**, which is **over 3× the VRAM** of a 4090. On Runpod, an A100 80GB runs about **$0.79/hr** on secure on-demand instances [6]. The A100's raw compute (FP32) is lower than a 4090, but it excels at mixed-precision training with Tensor Cores. If your workload is heavy on FP16/BF16 (which many AI training workloads are), an A100 can perform quite well. The large memory buffer means you can train **very large models or batches** without running OOM. *Code/optimization compatibility:* Your existing CUDA code will run on A100 without modification. In fact, frameworks like PyTorch will automatically use A100's tensor cores for acceleration (no code changes needed). You might find that you no longer need aggressive memory optimizations (like gradient checkpointing or offloading) when you have 80 GB available – simplifying your training script. Keep in mind the **cost vs. benefit**: an A100 is roughly 4× the cost of a 4090 per hour [7] [8], but if it lets you train in fewer steps or with less pipeline overhead, it could be worth it. This is a good choice if **48 GB might still be insufficient** for your model and you want a big memory headroom.

- **NVIDIA** H100 (80 GB) ***– 80 GB VRAM***: The H100 is NVIDIA's latest Hopper-generation GPU, also with **80 GB VRAM** (or up to 94 GB in the special H100 NVL variant). It is one of the **fastest GPUs available** for AI, offering newer features like FP8 precision and even higher throughput on matrix operations. On Runpod, H100 (80GB) is around **$1.35/hr** on-demand [9] – significantly more expensive than A100 or other options, but also extremely powerful. In many training tasks, an H100 can outperform an A100 or 4090, so if **time is critical** and you don't mind the cost, this could be worthwhile. *Code compatibility:* Your code will run on H100 with no changes (it's backward-compatible with CUDA). To truly leverage its capabilities, you could use libraries or settings that enable FP8 training or other Hopper-optimized features, but that's optional. Even without that, you'll benefit from faster throughput due to more Tensor Cores and higher FLOPs. Given the high price, H100 is usually recommended if you **need both high VRAM and maximum speed** – possibly overkill if you're mainly concerned about VRAM capacity. (Runpod's own comparison noted that even **on Runpod's affordable rates, H100 ~ $2.79/hr vs A100 80GB ~$1.19/hr on secure cloud** [10], so you pay a premium for the top-tier GPU.)

- **NVIDIA** RTX 5090 (32 GB) ***– 32 GB VRAM***: *(Emerging option)* The RTX 5090 (Blackwell architecture) is a next-generation consumer GPU that offers **32 GB** of VRAM – a 33% increase over the 4090's memory. This GPU is expected to have even higher compute performance than the 4090 (being the newer generation). Runpod has listed the RTX 5090 (32 GB) in their GPU catalog [11], so it may be available or coming soon on the platform. Pricing isn't officially listed on Runpod's site yet for the 5090, but it will likely fall somewhere between the 4090 and the pro 48GB cards. The 5090 could be a good middle-ground if your memory needs are only slightly above 24 GB – it gives a bit more headroom (32 GB) and faster speed, without jumping all the way to 48 GB. *Code compatibility:* Being an NVIDIA CUDA GPU, it should run your existing code just like the 4090 (no changes needed). This might be a viable choice if you find 24 GB is just *barely* insufficient – however, if you're going to invest in switching GPUs, the 48 GB options provide a bigger safety margin for memory-heavy workloads.

Summary of Options: For a **balanced upgrade**, the **RTX 6000 Ada 48GB** is a strong candidate – double VRAM while preserving performance, at roughly double the cost [7]. If budget is a big concern and you can handle a slower pace, the **RTX A6000 48GB** gives the same VRAM for a lower hourly rate [3], trading off some speed. If your job truly demands **far more memory**, consider the **A100 80GB** (for a large jump in VRAM at moderate cost increase) or the **H100 80GB** (if top performance is worth the high cost). All of these GPUs are **compatible with your existing code/optimizations** – no fundamental code changes should be required when switching from a 4090, aside from possibly adjusting any device-specific flags or enabling new features if you want. The main differences will be in **performance and how much batch/model you can fit** in memory, not in code functionality.

On-Demand vs. Spot Instances on Runpod

You expressed concern about **spot instances** (preemptible community GPUs) potentially interrupting your job and causing progress loss. Here's what you should know:

- **On-Demand (Secure Cloud) Instances:** On Runpod, the Secure Cloud instances are akin to on-demand, dedicated GPUs in reliable data centers [12] [13]. These will **not** cut out mid-run – you have the GPU as long as you rent it. The trade-off is higher cost. (Community vs. Secure can also influence cost: Community Cloud nodes are cheaper because they're often third-party/peer providers, but they should still be stable unless marked as spot [14].)

- **Spot (Community) Instances:** Runpod's Community Cloud (and any labeled "spot" instances) are cheaper, sometimes **40–70% lower cost** than on-demand [15]. However, they *can* be **preempted** – meaning the provider might reclaim the GPU, terminating your pod. If that happens, your process would be killed unless you've engineered a migration or checkpoint mechanism. *Runpod does have a feature to mitigate this:* they mention a **"spot-to-on-demand fallback"** which can automatically migrate your workload to an on-demand machine if a spot instance is about to shut down [16]. This means in theory, you wouldn't lose all progress; the platform tries to keep your pod running by switching to a paid instance if a spot one goes away. Even with this, you should assume that **spot instances can interrupt** and plan accordingly (e.g. save checkpoints regularly).

Recommendation on spot vs on-demand: If your workload is long-running and **cannot easily handle interruptions**, it's safer to choose **on-demand instances**. This guarantees you won't be cut off mid-process (so you won't "lose money on the way" with a failed run). If you do want to take advantage of lower cost **spot pricing**, make sure your code periodically saves state (so you can resume after an interruption) and be prepared for possible restarts. Many batch training jobs can utilize spot instances effectively by checkpointing models and data often [17]. But for a sensitive or very lengthy job, the peace of mind of an on-demand instance may be worth the extra cost. Given your caution about wasted cost, you'll likely prefer sticking with **on-demand (Secure Cloud) on Runpod** unless you've implemented a robust resume strategy.

Conclusion and Guidance

In summary, upgrading to a GPU with more VRAM on Runpod could remove the memory bottleneck you're facing with the RTX 4090:

- If you want **minimal changes and strong performance**, go with a **48 GB GPU** (RTX 6000 Ada for maximum speed, or RTX A6000 for cost efficiency) to double your available VRAM. This is often enough to solve memory issues while only doubling or less your hourly cost [1] [3].
- For **even heavier memory needs**, an **80 GB A100** offers a huge VRAM boost at ~4× cost of a 4090 [8], and an **H100** gives top performance at a higher price premium [9]. These would ensure memory is no longer a limiting factor at all, though you pay more for the capability.
- You **do not need to modify your code** when switching to any of these NVIDIA GPUs – they all support CUDA and standard deep learning frameworks. At most, you might disable some memory-saving tricks (if you had enabled them for the 4090) to take advantage of the additional VRAM.
- Given your cost considerations, you might start by trying an **on-demand RTX 6000 Ada 48GB** instance. It's a balanced choice: similar speed to your 4090, double the memory, and a fair hourly

rate. Monitor the GPU memory usage; if you find you still need more headroom, you can then step up to an 80 GB GPU.
- Finally, prefer **on-demand instances** for stability. Use spot instances only if you've accounted for possible interruptions (with frequent checkpoints or shorter tasks) [17]. This way, you won't risk losing progress and money unexpectedly.

By choosing the right GPU and pricing model, you can accelerate your workload without hitting memory limits, while keeping costs in check. Good luck with your runs!

Sources:

- Runpod GPU Catalog – *Available GPU types and VRAM* [18] [19]; *Pricing for various GPUs on Runpod* [20] [21] [3].
- Runpod Pricing Blog – *Example rates on Runpod vs others (A100, H100, 4090)* [10]; *Cost-saving strategies (GPU selection, spot vs on-demand)* [22] [4].

---

[1] [2] [3] [6] [7] [8] [9] [16] [20] [21] RunPod GPU Pricing & Specs | ComputePrices.com
https://computeprices.com/providers/runpod

[4] [5] [14] [15] [17] [22] How can I reduce cloud GPU expenses without sacrificing performance in AI workloads?
https://www.runpod.io/articles/guides/reduce-cloud-gpu-expenses-without-sacrificing-performance

[10] Runpod vs. CoreWeave: Which Cloud GPU Platform Is Best for AI Image Generation?
https://www.runpod.io/articles/comparison/runpod-vs-coreweave-which-cloud-gpu-platform-is-best-for-ai-image-generation

[11] [18] [19] GPU types - Runpod Documentation
https://docs.runpod.io/references/gpu-types

[12] What is the difference between secure cloud and Community Cloud?
https://www.answeroverflow.com/m/1209624710896295996

[13] What to Look for in Secure Cloud Platforms for Hosting AI Models
https://www.runpod.io/articles/guides/secure-ai-cloud-platforms