RTX 4090 · Inference Optimization

EXP-ISODICE-001
REV I · 2026-06-10

iso-Dice · kidney-tumor CT segmentation (KiTS) · pre-trained nnU-Net (frozen) · training-free inference settings · vs NVIDIA L4 cloud baseline · 10 held-out cases (480–489)

1.0 GOAL & SETUP

WHAT & HOW

Goal — measure how far training-free inference settings reduce the energy and runtime of a fixed, pre-trained kidney-tumor segmentation network (nnU-Net) on a desktop-class RTX 4090, at what cost to segmentation quality, and how the optimized RTX 4090 compares to the same workload on a cost-efficient cloud GPU (NVIDIA L4). Setup — settings are applied cumulatively (numerical precision → patch overlap → test-time augmentation), each run over the same 10 held-out KiTS cases (480–489) on both cards. Quality is measured per region (kidney, masses, tumor) against a pre-registered clinical-acceptability threshold; the tumor region is the limiting one. Energy is each GPU's on-chip NVIDIA hardware counter (NVML), measured above idle; an independent Zeus counter agreed to within 0.01 %.

KEY TERMS

RTX 4090 / L4 — a desktop GPU that can be self-hosted vs a low-power GPU rented in the cloud (the deployment baseline here).

fp16 / fp32 — half- vs single-precision arithmetic. Lower precision runs faster and uses less energy but can reduce numerical accuracy.

TTA (test-time augmentation) — running the model on flipped/rotated copies and averaging: more accurate, several times more compute.

Patch step size (0.50 / 0.75) — how far the sliding window advances each move, as a fraction of the patch. Step 0.50 = 50 % overlap between patches; step 0.75 = only 25 % overlap. A larger step = fewer patches = faster and lower-energy, but coarser at patch edges.

Dice — overlap between the predicted and reference segmentation (0–1; higher is better).

Net energy — GPU energy above idle, from the on-chip NVIDIA counter (NVML), reported per case.

2.0 OPTIMIZATION SEQUENCE — MEASURED ON THE RTX 4090

10 HELD-OUT CASES · NVML COUNTER

PARAMETER	B0 baseline fp32·TTA·.50	O1 half prec. fp16	O3 reduced fp16·.75·noTTA	Δ O3÷B0
SPEED & ENERGY				
Wall time (s / case) perf_counter around the per-case sliding-window inference.	187.6	123.4	42.4	0.23×
GPU energy (kJ / case) On-die NVML counter, end-minus-start delta. Hardware truth.	49.4	29.3	3.13	0.063×
Avg GPU power (W) Mean of the NVML power samples over the run.	263	237	74	0.28×
Speed-up vs B0 Baseline wall time ÷ this setting's wall time.	1.0×	1.5×	4.4×	—
Energy reduction vs B0 Baseline energy ÷ this setting's energy.	1.0×	1.7×	15.8×	—
SEGMENTATION QUALITY — DICE, HELD-OUT				
Tumor Dice Limiting region. Overlap vs reference annotation (0–1).	0.887	0.887	0.780	-0.107
Kidney Dice Whole-kidney region Dice.	0.976	0.976	0.952	-0.024
Masses Dice Kidney-masses region Dice.	0.902	0.902	0.798	-0.104
RESOURCES				
Peak VRAM (MB) Max NVML framebuffer-used over the run.	4690	3123	3123	0.67×

TBL-1. RTX 4090, 10 held-out KiTS cases (480–489), NVML hardware counter. Settings are cumulative. O1 (half precision) is essentially lossless (Δ tumor Dice +0.0001) at 1.7× lower energy and 1.5× faster. O3 lowers energy 15.8× and runtime 4.4× but reduces tumor Dice by 0.107; whether that is acceptable is decided by the pre-registered clinical threshold. O2 (8-bit) was not run; O4 (compiled runtime) is covered in §4. Here B0 is the 4090's own on-card baseline; the comparison against the separate cloud L4 baseline is in §3.

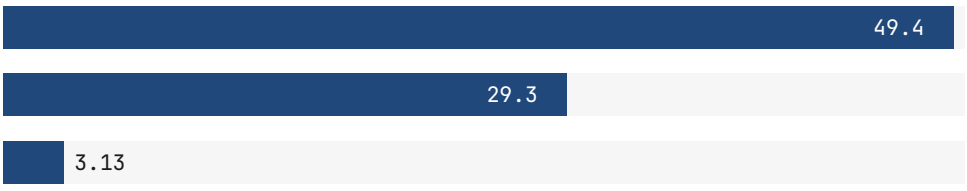

FIG-1. Net GPU energy per case along the sequence (RTX 4090). Energy falls faster than runtime because each step also lowers average power draw.

3.0 RTX 4090 RUNS vs THE L4 CLOUD BASELINE

L4 BASELINE = REFERENCE · IMPROVEMENT PER 4090 RUN

CONFIGURATION	s/CASE	kJ/CASE	TUMOR DICE	SPEED vs L4	ENERGY vs L4
BASELINE — CLOUD NVIDIA L4					
L4 baseline (reference) Cloud L4, fp32 · TTA · 0.50 — the deployment baseline everything below is measured against.	499	34.1	0.887	1.0× (ref)	1.0× (ref)
DESKTOP RTX 4090 — SAME MODEL, SAME 10 CASES					
4090 · B0 same settings fp32 · TTA · 0.50. Hardware change only — no optimization yet.	187.6	49.4	0.887	2.7× faster	1.45× more
4090 · O1 half precision fp16. Identical Dice (lossless).	123.4	29.3	0.887	4.0× faster	1.16× less
4090 · O3 reduced fp16 · 0.75 · no TTA. Tumor Dice -0.107.	42.4	3.13	0.780	11.8× faster	10.9× less
4090 · O4 + TensorRT O3 settings + native TRT engine (§4). Same Dice; wall unchanged (TRT is per-patch), ~15 % less energy, ~40 % less VRAM.	≈42.4	2.65	0.780	11.8× faster	12.9× less

TBL-2. Top row is the cloud L4 baseline (reference, 1.0×); each RTX 4090 row below shows its improvement over that baseline. Moving to the 4090 at the same settings (B0) is 2.7× faster but uses 1.45× more energy; adding half precision (O1) makes it 4.0× faster and 1.16× less energy than the L4 baseline at identical Dice; O3 is 11.8× faster and 10.9× less energy but lowers tumor Dice by 0.107. The O4 row stacks the native TensorRT engine on O3 (detail in §4): TensorRT speeds the per-patch forward but not the whole scan, so wall time is unchanged, while per-case energy drops a further ~15 % to 2.65 kJ — 12.9× below the L4 baseline — at the same 0.780 Dice and ~40 % less VRAM. (B0/O1/O3 are the 10-case sweep; the O4 energy is the 5-case A/B of TBL-4.) Note — setting-for-setting (both cards at fp16) the L4 still uses 1.2–1.45× less energy than the 4090; the 4090's net win comes from pairing its higher speed with the optimizations.

OBSERVATION

Against the cloud L4 baseline (top row), the RTX 4090 is faster at every configuration, and once half precision is enabled (O1) it is also lower-energy — 4.0× faster and 1.16× less energy at identical accuracy. The largest-efficiency run (O3) is 11.8× faster and 10.9× lower energy, at a 0.107 reduction in tumor Dice. Both cards produce identical Dice, so the differences are efficiency only. (At matched settings the L4 remains the more energy-efficient card; the 4090's net advantage over the baseline comes from speed combined with the optimizations.)

4.0 COMPILED RUNTIME (TensorRT)

RTX 4090 · PER-PATCH 128^3

BUILD	PRECISION	ms/PATCH	vs autocast fp16	ACCURACY
eager (PyTorch)	fp32	32.8	1.75× slower	correct (reference)
eager (PyTorch)	autocast fp16	18.8	1.00× (reference)	correct
torch-tensorrt	fp16	13.4	1.40× faster	incorrect (Dice 0)
native ONNX→TRT	fp16	9.6 (fastest)	1.96× faster	correct (Dice = eager)
native ONNX→TRT	fp32	29.0	1.54× slower	correct

TBL-3. Per-patch latency on the RTX 4090 (mean over repeated forward passes on a fixed 128^3 patch); the autocast-fp16 path is the reference, since it is what O1 and O3 already use. Correction to the earlier finding: the fp16 accuracy collapse (Dice 0, a real patch labelled entirely as background) occurs only when the engine is built with torch-tensorrt, which forces fp16 on the normalization internals (its build warned 16 overflow / 70 subnormal layers). Building the engine natively from an ONNX export avoids this — InstanceNorm runs as a numerically-safe TensorRT plugin — so the native fp16 engine is both the fastest measured (9.6 ms, 1.96× faster than autocast) and correct. A precision-constrained build (pinning normalization/softmax to fp32) was also tested but pinned 0 layers, since the exported graph exposes no such layers to constrain: plain native fp16 needs no precision constraints. End-to-end validation — full sliding-window Dice (cases 480 / 481 / 482), native-TRT-fp16 vs eager-fp32: tumor 0.789/0.789, 0.958/0.958, 0.728/0.726; kidney-and-masses 0.977/0.977, 0.971/0.971, 0.922/0.922 — lossless to within 0.001.

PER CASE (step 0.75 · no TTA · 5 cases)	autocast fp16	native TRT fp16	TRT effect
Wall time (s / case)	58.0	56.5	1.03× (~flat)
GPU energy (kJ / case)	3.12	2.65	0.85× (-15 %)
Peak VRAM (MB)	3935	2343	0.60× (-40 %)
Tumor Dice (mean)	0.706	0.706	identical

TBL-4. Autocast vs native-TRT fp16 measured in the same pipeline (NVML counter, 5 held-out cases). The 1.96× per-patch speed-up (TBL-3) does not carry over to per-case wall time — the network forward is only a small slice of a whole scan, and the rest (preprocessing, gaussian-weighted tiling, resampling, NIfTI export) is identical for both paths. TRT's real per-case payoff is a modest ~15 % energy reduction and ~40 % lower peak VRAM, losslessly; per-case throughput is essentially unchanged. (Caveat: combining the native engine with test-time augmentation is currently unreliable and was excluded.)

5.0 FINDINGS

SUMMARY OF RESULTS

O1 — half precision	Essentially lossless: identical Dice in every region (Δ tumor +0.0001) at 1.7× lower energy (49.4 → 29.3 kJ/case) and 1.5× faster. Recommended as a default.
O3 — reduced overlap, no TTA	Widens the sliding step to 0.75 (25 % patch overlap, down from 50 % at step 0.50) and removes TTA — the largest efficiency gain: 15.8× less energy (49.4 → 3.13 kJ/case) and 4.4× faster, with average power down to 74 W, but tumor Dice falls from 0.887 to 0.780 (-0.107; masses 0.902→0.798). Appropriate only within the clinical-acceptability threshold.
RTX 4090 vs L4	Consistent across all settings: the 4090 is 1.6–2.7× faster per scan, the L4 uses 1.2–1.45× less energy. The 4090 suits turnaround time, the L4 suits energy and operating cost; segmentation quality is the same on either card.
O4 — compiled runtime	A native ONNX→TensorRT fp16 engine is 1.96× faster per patch and lossless — the earlier Dice-0 failure was a torch-

tensorrt artifact, not TensorRT, and no precision constraints were needed. But end-to-end (TBL-4) the per-patch win does not reach per-case wall time (1.03×, flat); the practical per-case payoff is ~15 % less GPU energy and ~40 % lower VRAM, losslessly. A small, real efficiency gain — not a throughput speed-up.

SUMMARY

On the RTX 4090, half precision (O1) is the clear default — it halves energy (29.3 vs 49.4 kJ/case) with no measurable accuracy cost. Reducing patch overlap and removing test-time augmentation (O3) gives the largest gain, 15.8× less energy (3.13 kJ/case), but lowers tumor Dice from 0.887 to 0.780 (-0.107) and is appropriate only within the clinical-acceptability threshold. Against the L4 cloud baseline on the identical workload, the 4090 is 1.6–2.7× faster per scan but uses 1.2–1.45× more energy at every setting, because the L4 draws far less power. Graph compilation helps modestly: a native ONNX→TensorRT fp16 engine is ~2× faster per patch and lossless (correcting the earlier “no gain” result), but end-to-end that buys only ~15 % less energy and ~40 % lower VRAM per case at essentially unchanged wall time (§4), because the network forward is only a small part of each scan. In absolute terms, the maximum-efficiency configuration scores tumor Dice 0.780 versus the original 0.887; the lossless path (O1) keeps the original 0.887. Tumor Dice matches across both cards to within 0.001, confirming the optimizations and the hardware affect only efficiency, not what the model predicts. All figures are hardware-counter measurements over the same 10 held-out cases.